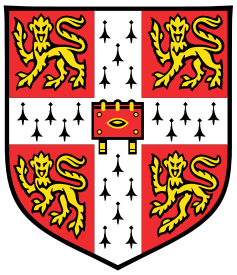
# A Multiplicative Regularisation for Inverse Problems



**Yujun Qiao**

Department of Applied Mathematics and Theoretical Physics
University of Cambridge

This dissertation is submitted for the degree of
*Doctor of Philosophy*

Trinity Hall College February 2018

I would like to dedicate this thesis to my loving parents Jihong Qiao and Baoqian Li.

# Declaration

I hereby declare that this thesis is the result of my own work and includes nothing which is the outcome of work done in collaboration except as declared in the Acknowledgements and specified in the text. It is not substantially the same as any that I have submitted, or, is being concurrently submitted for a degree or diploma or other qualification at the University of Cambridge or any other University or similar institution except as declared in the Acknowledgements and specified in the text. I further state that no substantial part of my thesis has already been submitted, or, is being concurrently submitted for any such degree, diploma or other qualification at the University of Cambridge or any other University or similar institution except as declared in the Acknowledgements and specified in the text. It does not exceed the prescribed word limit for the relevant Degree Committee.

Yujun Qiao

# Acknowledgements

Firstly I would like to give an enormous thank you to my supervisor, Dr Orsola Rath-Spivack, not only for the intelligence and effort she has put into my PhD project, but also for her guidance and advice throughout all my years being in Cambridge since an undergraduate student. She has nurtured me greatly as a PhD supervisor as well as a Director of Studies, and lead my career path along mathematics. I would not be able to achieve wherever I get to today without all her kind support.

Secondly, I would like to thank my advisor Dr Carola Schölieb and Dr Mark Spivack, who were always able to offer help when I need it, and making sure that I was on track for all the progress.

I owe great thanks to a list of people either from my department or other research institute: Dr Yiqiu Dong, Prof Mila Nikolova, Dr Martin Benning, Prof Paul Barbone, Dr Zhu Huang, Dr Stephen Eglen, Mr Kai Cao... Working through the PhD project is a long and sometimes daunting process, and I consider myself very lucky meeting these people along the way. They helped me with various aspect of my project: from thesis content to Latex referencing, from technical discussion to mental encouragement. The Cambridge imaging analysis group (CIA) offers inspiring group talks weekly, and I benefited greatly by participating.

I would also like to take this opportunity to thank my sponsors, Cambridge Overseas Trust and my college Trinity Hall, for both offering me scholarships and supported me financially for this project. My experience as a PhD student in Cambridge is invaluable, and it plays a crucial role in my personal development alongside my academic life. I am very grateful for my funding body, and would be happy to contribute in the future as I can.

Last but not the least, I would like to thank my families, who has supported me for my entire study in the UK and in Cambridge. Taking undergraduate study in the UK after finishing my secondary school in China was a big decision for me as well as for my entire family. Looking back into it today I just cannot appreciate this opportunity more. I am deeply

grateful to all my families who helped me realised this dream and set off my future career.

# Abstract

**Thesis title: A Multiplicative Regularisation for Inverse Problems**
**Author: Yujun Qiao**
**Department of Applied Mathematics and Theoretical Physics**

This thesis considers self-adaptive regularisation methods, focusing particularly on new, multiplicative methods, in which the cost functional is constructed as a product of two terms, rather than the more usual sum of a fidelity term and a regularisation term.

By re-formulating the multiplicative regularisation model in the framework of the alternating minimisation algorithm, we were able to obtain a series of rigorous theoretical results, as well as formulating a number of new models in both multiplicative and additive form.

The first two chapters of my thesis set the scene of my research. Chapter 1 gives a general review of the field of inverse problems and common regularisation strategies, while Chapter 2 provides relevant technical details as mathematical preliminaries. The multiplicative regularisation model by Abubakar *et al* (2004) falls into the category of self-adaptive methods, where the regularisation strength is automatically adjusted in the model. By investigating the model and implementing it on various examples, I demonstrated its power for deblurring piecewise constant images with the presence of noise with high amplitude and various distributions (Chapter 3). I also discovered a possible improvement of this model by the introduction an extra parameter $\mu$, and came up with a formula to determine its most appropriate value in a straightforward manner. The derivation and numerical validation or this formula is presented in Chapter 4. This parameter $\mu$ supplements Abubakar's multiplicative method, and plays an important role in the model: it enables the multiplicative model to reach its full potential, without adding any significant effort in parameter tuning.

Despite its numerical strength, there are barely any theoretical results regarding the multiplicative type of regularisation, which motivates me to carry out further research in this aspect. Inspired by Charbonnier *et al* (1997) who provided an additive model with

regularisation strength spatially controlled by a sequence of self-adapted weight functions $b_n$, I re-formulated the multiplicative regularisation model in the framework of alternating minimisation algorithm. This results in a series of new models of the multiplicative type. In Chapter 5 I presented two new models MMR and MSSP equipped with two-step and three-step alternating minimization algorithm respectively. The scaling parameter $\delta$ is fixed in the former model while it is self-adaptive based on an additional recurrence relation in the latter model. In both models, the objective cost functional $C_n$ is monotonically decreasing and convergent, while the image intensity $u_n$ exhibits semi-convergence nature. Both models are capable of incorporating different potential functions in the objective cost functional, and require no extra tuning parameter $\mu$ in the algorithm. Numerically they exhibit similar behaviours as Abubakar's multiplicative method in terms of high noise level tolerance and robustness over different noise distributions.

In Chapter 6 I presented a third, enhanced multiplicative model (EMM), which employs not only a three-step minimisation with self-adaptive weight function $b_n$ and scaling parameter $\delta_n$, but also the same augmented recurrence relation as discussed in Chapter 4 with steering parameter $\mu$. This model leads to promising results both theoretically and numerically. It is a novel approach with enhanced performance exceeding all the multiplicative type of models presented in this dissertation.

**Keywords**: Self-adaptive regularisation, multiplicative regularisation, deblurring, denoising, regularisation with spatial dependence, piecewise constant image, non-convex objective functional, conjugate gradient method, alternating minimisation algorithm, optimisation, gradient descent.

# A Multiplicative Regularisation for Inverse Problems

Yujun Qiao
Department of Applied Mathematics and Theoretical Physics
University of Cambridge

Feburary 8, 2018

# Contents

# List of Notation and Acronyms

| | |
|---|---|
| $\mathcal{X}$ | A normed space, usually a Hilbert space. |
| $A$ | A linear operator acting between two Hilbert spaces $\mathcal{X}$ and $\mathcal{Y}$, unless otherwise specified. |
| $F$ | A general (usually nonlinear) operator acting between two Hilbert spaces $\mathcal{X}$ and $\mathcal{Y}$, unless otherwise specified. |
| $A^*$ | adjoint operator of $A$. |
| $A^\dagger$ | The Moore-Penrose generalised inverse of $A$. |
| $\mathcal{D}(A)$ | The domain of $A$. |
| $\mathcal{R}(A)$ | The range of $A$. |
| $\mathcal{N}(A)$ | The null space of $A$. |
| $\overline{\mathcal{R}(A)}$ | The closure of $\mathcal{R}(A)$. |
| $\mathcal{R}(A)^\perp$ | The orthogonal complement of $\mathcal{R}(A)$. |
| $y$ | Exact data. |
| $y^\epsilon$ | Noisy data. |
| $\|\| \cdot \|\|$ | The $L^2$ norm, unless otherwise stated in the context. |
| $\langle \cdot, \cdot \rangle$ | inner product. |
| $\delta F(u)[v]$ | Directional derivative of $F$ at $u$ in the direction of $v$. |
| $TV[u]$ | Total variation of image intensity $u$. |
| $BV$ | Space of functions of bounded variation. |
| $\Sigma$ | Diagonal matrix with singular values. |
| $U$ | left singular matrix. |
| $u_i$ | left singular function or vector. |
| $V$ | right singular matrix. |
| $v_i$ | right singular function or vector. |
| $\sigma_i$ | $i$th singular eigenvalues. |
| $h$ | Mesh size. |

| | |
|---|---|
| $N$ | Number of pixels in $x$ and $y$ direction . |
| $\Omega$ | Discretised image domain containing $N \times N$ pixels. |
| $u$ | Image intensity, dimension $N \times N$. |
| $u_{i,j}$ | Element-wise representation of image intensity. |
| $u_n$ | $n$th 2D image intensity, dimension $N \times N$. |
| $\mathbf{u}_n$ | Reshaped $n$th 2D image intensity into $1 \times N^2$ array. |
| $u_n^m$ | Image intensity at the $m$th inner loop of the $n$th outer loop, with dimension $N \times N$. |
| $M_n$ | Total number of inner loop iterations conducted at the $n$th outer loop |
| $M_{outer}$ | Maximum number of iteration conducted in the outer loop. |
| $M_{inner}$ | Maximum number of iteration conducted in the inner loop. |
| $u^{exact}$ | intensity of the original image, dimension $N \times N$. |
| $\epsilon$ | noise level, a scalar. |
| $f$ | Blurred image, dimension $N \times N$. |
| $\eta$ | Noise component on the right hand with zero mean, dimension $N \times N$. |
| $f^\epsilon$ | Corrupted image (blurred and noisy), dimension $N \times N$. |
| $f_{i,j}^\epsilon$ | Element-wise representation of the corrupted image. |
| $\mathbf{f}^\epsilon$ | Reshaped 2D corrupted image into $1 \times N^2$ array. |
| $\mathcal{K}$ | Blurring operator, apply to 2D $N \times N$ matrices. |
| $\mathbf{K}$ | Reshaped blurring operator into $N^2 \times N^2$ matrix, apply to $1 \times N^2$ array. |
| $\delta_n^2$ | $n$th steering parameter in the in the multiplicative regularisation model, a scalar. |
| $D_x$, $D_y$ | Finite difference operator in $x$ and $y$ direction. |
| $\mathbf{D}$ | Reshaped finite difference operator into $N^2 \times N^2$ matrix, apply to $1 \times N^2$ array. |
| $\Omega_n$ | Set contains all elements whose finite differences $(D_x u_n)_{i,j}^2 + (D_y u_n)_{i,j}^2$ is larger than $\delta_n^2$. |
| $\Omega_n^c$ | Complement of $\Omega_n$ in image domain $\Omega$. |
| $C_n(u)$ | $n$th objective cost functional. |
| $F(u)$ | Data fidelity term. |
| $F_n^{multi}(u)$ | $n$th regularisation term of the multiplicative regularisation model. |

| | |
|---|---|
| $(b_n^2)$ | $n$th weight function in the multiplicative regularisation model. |
| $(b_n^2)_{i,j}$ | Matrix representation of the $n$th weight function, dimension $N \times N$. |
| $\mathbf{B}_n$ | $N^2 \times N^2$ diagonal matrix with diagonal element being $(b_n^2)_{i,j}$, $1 \leq i, j \leq N$. |
| $\alpha_n$ | $n$th step size in the multiplicative regularisation model, a scalar. |
| $\alpha_n^m$ | Step size at the $m$th inner loop of the $n$th outer loop, a scalar. |
| $g_n$ | $n$th gradient direction of the multiplicative regularisation model, dimension $N \times N$. |
| $\mathbf{g}_n$ | Reshaped 2D gradient direction into $1 \times N^2$ array. |
| $g_n^m$ | Gradient direction at the $m$th inner loop of the $n$th outer loop, dimension $N \times N$. |
| $v_n$ | $n$th search direction in the multiplicative regularisation model, dimension $N \times N$. |
| $\mathbf{v}_n$ | Reshaped 2D search direction into $1 \times N^2$ array. |
| $v_n^m$ | Search direction at the $m$th inner loop of the $n$th outer loop, dimension $N \times N$. |
| $\mu$ | Tuning parameter of the multiplicative regularisation model, a scalar. |
| $\lambda$ | Regularisation parameter of the additive regularisation model, a scalar. |
| SVD | Singular value decomposition |
| GCV | Generalised cross-validation. |
| CGM | Conjugate gradient method. |
| PSNR | Peak Signal-to-Noise Ratio. |
| SSIM | Structure SIMlarity Index. |
| MMR | Modified Multiplicative Regularisation. |
| MSSP | Model with Self-Adaptive Scaling Parameter. |
| EMM | Enhanced Multiplicative Method. |
| HL, GM, HS | Potential function in a specified type. |
| $\varphi$, $\psi$ | Potential function in the general form. |
| $\varphi_{pot}$, $\psi_{pot}$ | Potential function in the type of HL, GM or HS. |
| $C^{MMR}$ | MMR objective cost functional. |
| $C^{MSSP}$ | MSSP objective cost functional. |
| $C^{EMM}$ | EMM objective cost functional. |
| $C^{Charb}$ | Objective cost functional of the additive model by Charbonnier *et al.* |
| $C^{Abubakar}$ | Objective cost functional of the multiplicative model by Abubakar *et al.* |
| $C^{Add}$ | Objective cost functional of the additive counterpart corresponding to the multiplicative model as specified in the context. |

# Chapter 1

# Introduction

As one of the fastest growing areas in applied mathematics in the last few decades, inverse problems have been widely applied to science and industry. Inspired by Hadamard's work in 1902 and 1923, John pointed out in the 1960s that most of the mathematical models of inverse problems are not well-posed, with the major challenge being the lack of continuous dependence of the reconstructed solution over the initial data. In other words, a small perturbation in the initial data may cause a large difference in the final solution. This is undesirable, as in real world problems, only noisy data (*i.e.* exact data plus some small perturbation) is available. To address this difficulty, numerical methods known as regularisation methods are introduced to inverse problems. This dissertation investigates established and recent techniques for regularisation, focusing particularly on self-adapting regularisation method where an *a priori* regularisation parameter is not required in the model.

As J.B.Keller defined in 1976, two problems are called inverse to each other if the formulation of one problem involves another one [EHN96]. Usually the easier one or the one that is studied earlier is called the direct problem, while the other one is called the inverse problem. In real-world problems, the distinction between a direct and an inverse problem is clear in most cases. E.g., the prediction of the future behaviour of a physical system from its present state and the physical laws is called a direct problem, while the determination of the present state from future ob-

servations, or the identification of physical parameters from observation of evolution of the system may be its corresponding inverse problem.

The two major motivations to study an inverse problem are the following: Firstly, to find the past state or a parameter of a physical system; Secondly, to find out how to influence a system via its current state and parameter in order to steer it into a desired state in the future. Some typical examples of inverse problems include applications of sonar, radar, coastal evolution, mine detection, and image processing etc., where the technique are applied either for parameter recovery, or to restore state in the past and hence future prediction by reverting the process. In one sentence, inverse problems are concerned with determining causes for a desired or an observed effect [EHN96], which has been widely applied in the industry.

There is a vast literature on inverse and ill-posed problems, see, e.g. Engl, Hanke and Neubauer [EHN96]. In addition to the references quoted in later sections, we mention the following as a general reference:

- the following monographs: [Ang90, Blo81, Gla86, Pay75, PT87].
- the journals *Inverse Problems* (Institute of Physics Publ.), *Inverse Problems in Engineering* (Gordon & Breach), and *Journal of Inverse and Ill-Posed Problems* (VSP).

The remainder of this chapter introduces the main mathematical concepts and difficulties related to inverse problems, and gives a brief overview of the main methods that are used to find solutions. We provide a few simple examples and proofs illustrate the concepts and to help relate them to subsequent results in the dissertation.

## 1.1 Well-posedness

**Definition 1.1.1.** *According to Hadamard (in 1902, 1923) and John (1960), a problem is* ***well-posed*** *if*

- *There exists a solution to the problem (**existence**);*
- *There is at most one solution (**uniqueness**);*
- *The solution depends continuously on the data (**stability**).*

*Consequently, the violation of any of the three above leads to **ill-posedness** of an inverse problem.*

Consider the general framework of an inverse problem

$$F : \mathcal{X} \to \mathcal{Y},\ F(x) = y, \tag{1.1.1}$$

where $F$ denotes a linear or nonlinear operator between normed space $\mathcal{X}$ and $\mathcal{Y}$ (in this dissertation we shall consider only Hilbert space $\mathcal{X}$ and $\mathcal{Y}$, unless otherwise stated). With this setting, the above three requirements for well-posedness of a problem may be interpreted as follows:

- $F$ is **surjective**, *i.e.* $\forall y \in \mathcal{Y},\ \exists x \in \mathcal{X}$ *s.t.* $F(x) = y$. If $F$ is not surjective, then the equation $F(x) = y$ is not solvable for **all** $y \in \mathcal{Y}$ (**non-existence**).
- $F$ is **injective**, *i.e.* $\forall x_i, x_j \in \mathcal{X},\ x_i \neq x_j \implies F(x_i) \neq F(x_j)$. If $F$ is not injective, then the equation $F(x) = y$ may have **more than one** solution (**non-uniqueness**).
- $F^{-1}$ is **continuous**, *i.e.* for any sequence $\{x_n\}_{n\in\mathbb{N}}$ with $F(x_n) \to F(x^\dagger),\ n \to +\infty$, it follows that $x_n \to x^\dagger,\ n \to +\infty$. If this is not the case, then $\exists y\ , y' \in \mathcal{Y},\ ||y - y'|| \ll 1,\ ||F^{-1}(y) - F^{-1}(y')|| \gg 1$. In other words, small error in the initial data in the space of observation $\mathcal{Y}$ may lead to large errors in the reconstructed solution in the space $\mathcal{X}$ (**instability**).

The difficulties of solving inverse problems come from the lack of any of the three requirements for well-posedness above. Compared with the violation of the other

two requirements, the violation of the first criterion which causes the non-existence appears to be the least severe one. In practice, it is usually tackled by enlarging the class that the solution is allowed to be. Non-uniqueness refers to the fact that the object sometimes is not mapped in a one-to-one fashion by the given operator $F$, and usually in this case an extra condition is enforced in order to select a particular solution as the solution of the inverse problem. Instability is concerned with cases where a small perturbation in the initial data $y$ (which may be introduced due to experimental or numerical error in the process of data acquirement) in the application may cause arbitrarily large and spurious components in the solution. This problem is considered as the most important one from the practical point of view due to its wide appearance in real-world applications. It is also the aspect that is researched most heavily by applied mathematicians.

## 1.2 Linear Inverse Problems

In this section we introduce the canonical example of inverse problems on Hilbert spaces using the linear operator equation

$$Ax = y, \tag{1.2.1}$$

where $A : \mathcal{X} \to \mathcal{Y}$ is a bounded linear operator acting between two Hilbert spaces $\mathcal{X}$ and $\mathcal{Y}$. When the range of $A$ is closed (and it is never closed if $A$ is compact, unless $\mathcal{R}(A)$ is finite), the non-existence and non-uniqueness of the solution may be addressed by the concept of the Moore Penrose generalised inverse $A^\dagger$, which will be discussed in detail in Chapter 2. Simply speaking, $A^\dagger$ tackles the non-existence problem by including the orthogonal complement of the range of $A$, *i.e.* $\mathcal{R}(A)^\perp$, into its domain, and the non-uniqueness problem by using the fact that the operator $A\big|_{\mathcal{N}(A)^\perp}$ is injective. This can be illustrated most easily in finite dimensions, where the linear equation (1.2.1) becomes a linear system with $A : \mathbb{R}^m \to \mathbb{R}^n$ being a $m \times n$ matrix. If $m \neq n$, then a solution may not exist (over-determined or some under-determined

systems), or it may not be unique (under-determined system with an infinite number of solutions). In this finite case, when the solution does not exist, it is always possible to consider its "least squares solutions", which minimise the residue term $||Ax - y||^2$ and satisfy the normal equation $A^*Ax = A^*y$. Note that this solution does not have to be unique. In this case, we will consider the best-approximate solution, which is defined as the least squares solution with the smallest norm (minimum norm solution). The best-approximate solution is always a least square solution, and it is unique [EHN96]. In the case of infinite dimensional spaces, a least squares solution may not exist, but when it does then it is also a solution of the normal equation.

Failure to satisfy the third condition, which arises from the non-closedness of $\mathcal{R}(A)$ can occur only in infinite dimensional spaces and leads to the more complicated instability problem. To be more precise, when $dim(\mathcal{Y}) < \infty$, it is possible to decompose it as $\mathcal{Y} = \mathcal{R}(A) \oplus \mathcal{R}(A)^\perp$; while for $dim(\mathcal{Y}) = \infty$, we only have $\mathcal{Y} = \overline{\mathcal{R}(A)} \oplus \mathcal{R}(A)^\perp$. This means the Moore Penrose generalised inverse solution $x^\dagger = A^\dagger y$ exists for all $y \in \mathcal{Y} = \mathcal{D}(A^\dagger)$ in finite dimensional spaces, while this is not true in infinite dimensional spaces. In fact, it is possible to show that when both $\mathcal{X}$ and $\mathcal{Y}$ are infinite dimensional, the inverse operator of a completely continuous (compact and continuous) operator cannot be continuous or bounded [WYY10]. This leads to an unbounded generalised inverse $A^\dagger$, which will cause instability. A general remedy is to apply regularisation to operator $A$, which is essentially to approximate $A^\dagger$ by some other operator $R_\alpha$ with better stability properties. In the case of linear regularisation method, $R_\alpha$ is chosen to be a family of bounded linear operators defined on the same domain as $A^\dagger$ and converges to it pointwisely as $\alpha \to 0$. [BB18].

For finite dimensional problems the generalised inverse $A^\dagger$ is always bounded, which means the perturbation in observed data $y$ can only lead to finite perturbation in the solution $x$ [Han09]. Nevertheless, this finite perturbation depends on the eigenvalues of $A$ and can be arbitrarily large. Ill-conditioning, which occurs when $A$ is nearly singular (*i.e.* having eigenvalues with very small magnitude), is a major concern for finite dimensional problems. In Section 2.2 of this disserta-

tion, singular value decomposition (SVD) is used to construct a representation of the Moore-Penrose generalised inverse, and to illustrate how regularisation works in both finite and infinite dimensional spaces.

## 1.3 Regularisation Methods

In this section, I will give a simple example to demonstrate the importance of regularisation, and to show how it may be implemented in practice. The chosen example is an image deblurring problem in the presence of noise, solved using Tikhonov regularisation. The problem consists of recovering a simple 2-dimensional blocky image $u^{exact}(x_1, x_2)$ on a square domain $\Omega = [0,1] * [0,1]$, as given in Figure 1.1(a), from the blurred and noisy image $f^{\epsilon}$, as given in Figure 1.1(b) and defined by

$$\mathcal{K}u^{exact} + \eta = f^{\epsilon}, \tag{1.3.1}$$

where $\mathcal{K}$ is a given deblurring operator 3.1.1 with a choice of $\zeta = 0.1$, $\eta$ is known noise, and we have chosen

$$u^{exact}(x_1, x_2) = \begin{cases} 1, \text{when } x_1 \in [\frac{1}{4}, \frac{3}{4}], x_2 \in [\frac{1}{5}, \frac{2}{5}], \ or \ [\frac{3}{5}, \frac{4}{5}] \\ 0. \text{ Otherwise} \end{cases} \tag{1.3.2}$$

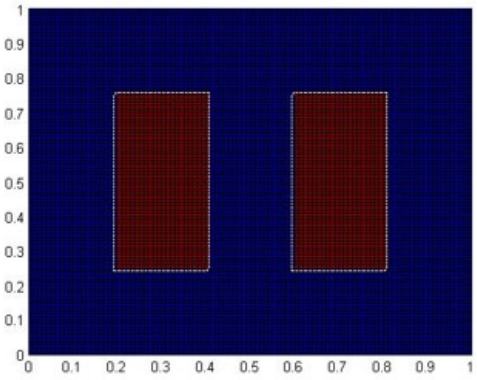
$\forall (x_1, x_2) \in \Omega = [0,1] * [0,1]$. Using Tikhonov regularisation (see Section 2.2 of this dissertation for details), this is equivalent to solving the following minimisation for the objective functional $J_{\alpha}$:

$$\min_{u \in C(\Omega)} J_{\alpha}(u(\mathbf{x})) =: \min_{u \in C(\Omega)} \{||\mathcal{K}u(\mathbf{x}) - f^{\epsilon}||^2 + \alpha ||u(\mathbf{x})||^2\}. \tag{1.3.3}$$

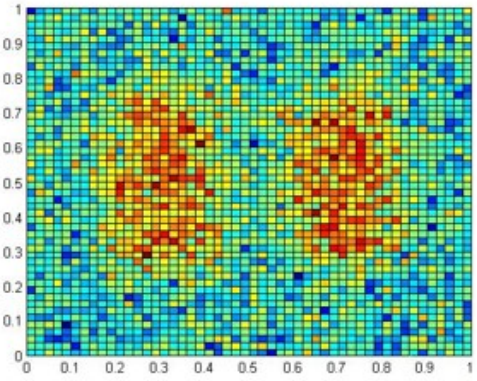
Here $\alpha$ is a "regularisation parameter", whose value needs to be pre-determined, and balances the relative contribution of the fidelity term $||\mathcal{K}u(\mathbf{x}) - f^{\epsilon}||^2$ and the

regularisation term $\alpha||u(\mathbf{x})|^2$. It is absolutely vital to make an appropriate choice of this parameter, as too small or too large a choice would cause under or over regularisation of the solution, and lead to poor reconstruction of the image. We solve (1.3.3) using a conjugate gradient method, for four different values of $\alpha$, one of which is a value $\overline{\alpha}$ chosen using an established parameter choice rule (the L-curve in the case), and the other values are chosen to illustrate over- and under-regularisation. The numerical results are shown in Fig 1.1(c)-(f): (c) shows the reconstructed image obtained with $\alpha = 0$. There is no regularisation in this case, and the solution shows the effects of numerical instability. In (d) we have chosen $\alpha < \overline{\alpha}$, showing the effect of under-regularisation; in (e) $\alpha = \overline{\alpha}$, which gives the best approximation of the original exact image; in (f) $\alpha > \overline{\alpha}$, which shows the effect of over-regularisation.

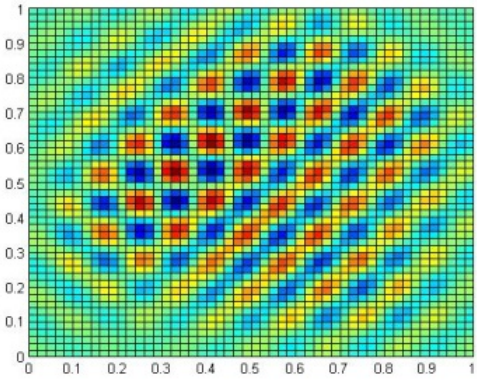
Although, as we have just seen with a simple example, it is absolutely essential to set an appropriate value for the regularisation parameter $\alpha$, this choice in practice is not easy to make. In Section 2.2 we will discuss various choice rules and their classification in detail. However, it is worth mentioning that all these techniques have their own shortcomings. For instance, the discrepancy principle requires a priori information of the noise level $\epsilon$, which is not necessarily always attainable in advance; the L-curve method relies on cross-validation and requires no knowledge of the input noise level, however, it requires either identification by trial and error, or further optimisation related to the curvature, both of which are computationally very expensive, and usually takes ten times more effort than the minimisation problem itself [AvdBHB03]. This motivates research areas such as iterative methods. Unlike traditional regularisation strategies such as the Tikhonov type which requires an a priori regularisation parameter $\alpha$ in the model, most iterative methods exhibit a "self-regularising" property in the sense that the early termination of the process regularises the system. To be more precise, the stopping index plays the role of the regularisation parameter, and the termination criterion of the algorithm helps to select its appropriate value [EHN96].



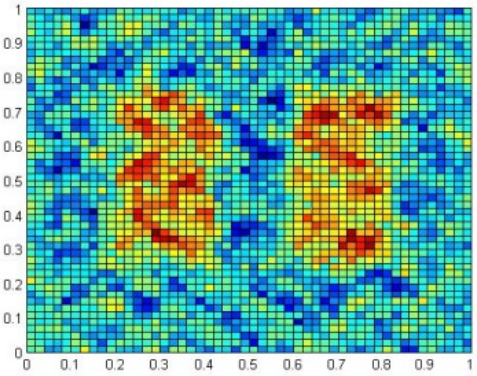
(a) exact image



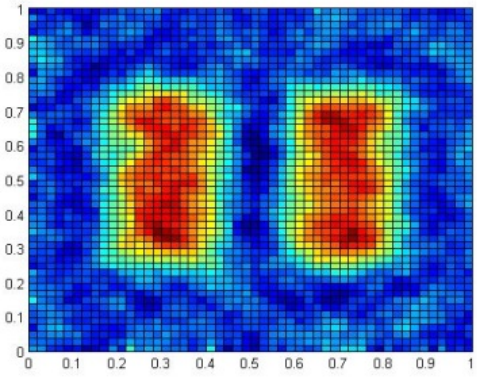
(b) corrupted image



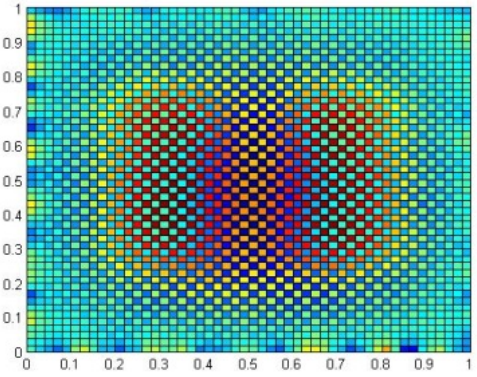
(c) $\alpha = 0$, no regularisation



(d) small value of $\alpha$, under-regularised



(e) appropriate value of $\alpha$, sensible reconstruction



(f) large value of $\alpha$, over-regularised

Figure 1.1: Reconstructed solutions with different parameter choices of $\alpha$

## 1.4 Nonlinear Inverse Problems

Compared to linear inverse problems which are researched relatively more extensively, there are various challenging aspects when the inverse problem is nonlinear. Consider the general setting

$$F : \mathcal{X} \to \mathcal{Y},\ F(x) = y, \tag{1.4.1}$$

where $F$ represents some nonlinear operator acting between Hilbert spaces $\mathcal{X}$ and $\mathcal{Y}$. A common, basic but reasonable theory is to assume that the operator $F$ is continuous as well as weakly sequentially closed (see, e.g. Engl,Hanke and Neubauer [EHN96], and Chapter 2.1.2 of this dissertation). In other words, we loosen the assumption and work in the setting of the weak topology in nonlinear inverse problems [EKN89]. This is often seen in the variational formulations of the nonlinear problems.

Note that although some of the concepts we have introduced in previous sections and illustrated with linear inverse problems (*e.g.* ill-posedness, regularisation, least-squares solution, Moore-Penrose generalised inverse etc.) may be naturally extended to nonlinear inverse problems with modification to some extent, singular value decomposition (SVD) is not one of them without the linearisation of the problem. This is because, the adjoint operator $F^*$ cannot be easily defined when $F$ is nonlinear, and even when it can be defined, in general it does not have the same properties as the adjoint of a linear operator [EK05]. When SVD is applied to nonlinear inverse problems, linearisation is usually required in the preparation. Actually, linearisation is one of the common strategies when dealing with nonlinear inverse problems. Although most of the inverse problems from practical applications are nonlinear, and their established theory is relatively incomplete, linear problems are still heavily investigated for research purpose, as they provide simpler forms as well as allowing some of the results to be extended to nonlinear problems.

This leads to one of the major difficulties when addressing nonlinear inverse problems. Apart from providing simplifications, linearisation may also introduce complexity in some cases. As it is shown in [EKN89], a nonlinear ill-posed problem may have a well-posed linearisation, and a well-posed nonlinear problem may also have ill-posed linearisation. This adds extra complexity and needs to be addressed in the linearisation process. Generally speaking, since the technique of linearisation is widely used in most of the numerical algorithms, it is also widely used in the application of nonlinear inverse problems.

Another common challenge to address for the nonlinear inverse problems when a solution is sought by minimisation, is the lack of convexity of the cost functional. This is usually not the case with linear inverse problems, whose fidelity is usually quadratic. With non-convex problems, the global minimiser of the objective cost functional cannot be simply characterised as the unique solution of the fixed point equation from the first order optimality condition, and global convergence is usually not guaranteed in many cases. This might jeopardise the robustness of the method, as not necessarily all the starting points will lead to satisfactory outcomes, which requires special care in these cases. In order to investigate function properties in the non-convex setting, Łojasiewicz [Loj63], [Loj93] and Kurdyka [Kur98] proposed the Łojasiewicz-Kurdyka property, allowing each bounded sequence generated by the algorithm to converge to a critical point of the function (see Chapter 2.4.2 of this dissertation). This idea has been taken further by Attouch [ABRS10], Nikolova and Tan [NT17] with an alternating proximal method for non-smooth and nonconvex functions. In the cases where a sufficient decrease condition is satisfied at each step, the algorithm may be proved convergent with finite path length.

As one of the most commonly used strategies, Tikhonov regularisation has been investigated and applied extensively in both linear and nonlinear cases (see, e.g., Kaltenbacher *et al* [KS08]). Despite its wide use for various ill-posed problems, Tikhonov regularisation has the drawback of oversmoothing the edges when applied to image applications. It can fail to preserve sharp features of the image, or cause

spurious oscillation [AvdBHB04]. In some extreme cases, detailed information in images can be lost entirely during the regularisation process [Lam01]. Other regularisation methods have been developed and successfully used. Iterative regularisation methods have been particularly useful in application to nonlinear inverse problems, where rigorous stability and convergence analysis has been carried out [KS08].

Generally speaking, iterative methods focus on finding the numerical solution of (1.4.1) using an iterative scheme in the form of

$$x_{k+1} = x_k + G_k(x_k, y),\ k = 1, 2, ... \tag{1.4.2}$$

with suitable choices of $G_k$ [KS08]. Under certain circumstances, the combination of scheme (1.4.2) plus an appropriate stopping index $k$ yield a stable solution for the nonlinear problem (1.4.1). There are pros and cons of this type of schemes: on the one hand, the simple and flexible form of (1.4.2) allows iterative schemes to accommodate nonlinear inverse problems in a more natural way; on the other hand, just as the tuning of regularisation parameters in classical regularisation methods such as the Tikhonov type, the choice of stopping rule for iterative schemes can be difficult and time-consuming. For certain iterative methods, the iterate $x_k$ initially tends to better and better approximations to the exact solution, then diverges from the exact solution and finally tends to a naive solution. This property is known as "semi-convergence", and requires an appropriate stopping rule for detecting the transition from convergence to divergence [Han09],[Han98],[EHN96]. Landweber iteration, one of the most well-known and widely applied iterative scheme exhibits semi-convergence. Its properties are well studied, and rigorous results are available ([KS08], [EHN96]). Details of this method are given in Section 2.4 of this dissertation. Apart from semi-convergence, the slow convergence rate is another major concern of this method.

## 1.5 Adaptive Regularisation

So far we have discussed regularisation methods in two different categories. The first one is formulated as a variational problem, where we assume a priori knowledge of an explicit regularisation parameter. The second one is the category of "iterative methods" equipped with a termination criterion, in which the stopping index plays the role of regularisation parameter. These two types of methods tackle the inverse problems from different perspective, and have their own strength and weaknesses. Variational regularisation methods are very well established, and allow variants suited to enhance particular features of a re-construction by using different kinds of penalty term in the objective cost functional to minimise; a major drawback in these methods is that the choice of the regularisation parameter may be computationally very expensive. Iterative methods are more suited to nonlinear problems, and have been shown in [BB18] to overcome the systematic reconstruction bias that can occur when using variational methods; however, their convergence rates may be slow.

For the recent few decades, researchers have tried to combine the merits and avoid the shortfalls of the two types of regularisation methods, by developing so-called "adaptive regularisation methods". These methods are "marriages" of the variational formulation and the iterative methods: An objective cost functional for optimisation is usually explicitly given in the model, while the regularisation effect is self-adjusting as the number of iteration goes along. These models often have high level of complexity, as enormous effort is dedicated to the design of the self-adaptation of the regularisation effect, which may be either controlled by a class of self-evolving regularisation parameters, or implicitly implemented in the algorithm.

This leads to the main topic of my thesis, the so-called multiplicative regularisation. This method appears to have regularising properties, and relies on minimising an objective cost functional constructed as the product instead of the sum of the data fidelity and the regularisation term. This causes the method to exhibit semi-convergence similarly to some iterative schemes. Research has shown [AvdBHB04],

[AvdBHB03] that this method is good at edge-preservation and hence works well for piecewise constant image reconstructions when solving image deblurring and denoising problems.

Before formally introducing the multiplicative regularisation model, Abubakar *et al* [AvdBHB04] tested its deblurring effect numerically on an inverse diffraction problem and demonstrated its robustness with a slightly different formulation of the regularisation term [AvdBHB03]. The calculation was complicated in this case due to a conjugate gradient method in the algorithm which requires a line search to update the search direction in every iteration. This results in multiple nested iterative loops, and takes large computation resources.

In the light of the adaptive regularisation model by Charbonnier *et al* [CBFAB97] where the regularisation term of the quadratic form can also achieve reasonable edge-preserving effects, Abubakar and van der Berg ([AvdB01]) replaced the regularisation term of the multiplicative method by a quadratic one, and demonstrated similar edge preservation effects numerically. In this case, the search direction of their algorithm had a closed form in each iteration, avoiding the need for a line search in every iteration. This means that the new type of regularisation can handle noisy data well without increasing the computational cost greatly. Based on these results, the formal approach of multiplicative regularisation was established by Abubakar *et al* in 2004 [AvdBHB04]. Since then it has been implemented for applications such as the inversion of seismic data [HAH09]. So far this method remains a very new one, and almost no theoretical results are available.

In this dissertation, I firstly tested Abubakar's model and demonstrated its effectiveness for deblurring piecewise constant images with in the presence of noise having high amplitude and various statistical distributions (Chapter 3). Then I improved this model by introducing an extra tuning parameter $\mu$, and provided a closed-form formula to determine its most appropriate value in a straightforward manner. This formula enables the multiplicative model to achieve its best image reconstruction

without adding any significant effort in parameter tuning, and is demonstrated to be effective and practical with various numerical examples (Chapter 4).

In Chapter 5 and 6 I constructed a series of multiplicative model MMR, MSSP and EMM in the framework of an alternating minimisation algorithm. These models inherit the strength of Abubakar's method, and are demonstrated to be robust over different noise distributions and high noise level, particularly in the case of piecewise constant images. Among these three new multiplicative models, MMR and MSSP exhibit better convergence results. By contrast with Abubakar's model, we are able to show that the objective cost functional $C_n$ is monotonically decreasing and convergent for these two models. In addition, both models require no extra tuning parameter $\mu$ in the algorithm, and may be generalised by incorporating different potential functions in the objective cost functional. Last but not least, as a novel enhanced method, EMM employs the same augmented recurrence relation as discussed in Chapter 4 with the pre-chosen parameter $\mu$, and numerically outperforms all the other multiplicative type of models presented in this thesis.

# Chapter 2

# Mathematical Preliminaries

## 2.1 Linear Analysis

### 2.1.1 Derivatives

Let $V$ denote a Banach space equipped with norm $||\cdot||_V$, and functional $F : V \to \mathbb{R}$. The analytical dual space of $V$ is denoted by $V^*$, *i.e.* the space of linear and continuous functional $T : V \to \mathbb{R}$. Following [Eva10], I will list a few definitions regarding differentiability and derivatives below.

**Definition 2.1.1.** *(Directional derivative) Let $F : V \to \mathbb{R}$. If for given $u, v \in V$, the limit*

$$\delta F(u)[v] := \lim_{h \to 0^+} \frac{F(u + hv) - F(u)}{h}$$

*exists, then $\delta F(u)[v]$ is called the directional derivative of $F$ at $u$ in the direction of $v$. If the limit exists for every $v \in V$, then $F$ is called directionally differentiable at $u$.*

A stronger notion of differentiability may be introduced in the cases where $F$ is everywhere directionally differentiable at $u$:

**Definition 2.1.2.** *(Gâteaux differentiability) Let $F : V \to \mathbb{R}$. If for a given $u \in V$ and every $v \in V$, the limit*

$$\delta F(u)[v] := \lim_{h \to 0^+} \frac{F(u + hv) - F(u)}{h}$$

*exists and $\delta F(u)[v] \in V^*$ is a linear and continuous functional with variable $v$, then $F$ is Gâteaux differentiable at $u$, and $\delta F(u)[v]$ is called the Gâteaux derivative of $F$ at $u$.*

Finally we introduce the notion of differentiability in terms of norm:

**Definition 2.1.3.** *(Fréchet differentiability) Let $F : V \to \mathbb{R}$. If $F$ is Gâteaux differentiable at $u \in V$ and satisfies in addition*

$$\lim_{||v||_V \to 0^+} \frac{\left|F(u + hv) - F(u) - \delta F(u)v\right|}{||v||_V} = 0,$$

*then $F$ is said to be Fréchet differentiable at $u$. In this case, the operator $\delta F(u)$ may be denoted by $\nabla F(u)$ and called the Fréchet derivative of $F$ at $u$.*

### 2.1.2 Weak Convergence

In this section I will present some definitions and results regarding weak topology, following work from Cheney [Che01]. I will work in the general setting of a normed linear space $X$ and its dual space $X^*$ (*i.e.* the space of linear and continuous functional $T : X \to \mathbb{R}$).

**Definition 2.1.4.** *(Weak convergence) A sequence $\{x_n\}_{n \in \mathbb{N}}$ in a normed linear space $X$ is said to **converge weakly** to an element $x \in X$ if $\phi(x_n) \to \phi(x)$ for every element $\phi \in X^*$. This may also be written as $x_n \rightharpoonup x$.*

The usual type of convergence may be termed **norm convergence** or **strong convergence**, which refers to $||x_n - x||_X \to 0$, and may be denoted as $x_n \to x$. Clearly the strong convergence implies the weak one, as each $\phi \in X^*$ is continuous.

Some properties regarding weak convergence are listed below:

**Lemma 2.1.5.** *A weakly convergent sequence is bounded.*

**Theorem 2.1.6.** *In a finite-dimensional normed linear space, weak and strong convergence coincide.*

The concept of **weakly sequentially closedness** and its properties are introduced as follows.

**Definition 2.1.7.** *A subset $U$ in a normed linear space $X$ is said to be* ***weakly sequentially closed*** *if the weak limit of any weakly convergent sequence in $U$ is also in $U$.*

A weakly sequentially closed set $D$ is necessarily closed in the norm topology, for if $x_n \in U$ and $x_n \to x$, it implies $x_n \rightharpoonup x$, hence $x \in U$ as $x_n$ is weak sequentially closed in $U$.

**Definition 2.1.8.** *A nonlinear operator $F : \mathcal{X} \to \mathcal{Y}$ acting between two Hilbert spaces $\mathcal{X}$ and $\mathcal{Y}$ is called weakly sequentially closed, if for any sequence $x_n \subset \mathcal{D}(F)$, where $\mathcal{D}(F) \subset \mathcal{X}$, $x_n \rightharpoonup x \in \mathcal{X}$ and $F(x_n) \rightharpoonup y \in \mathcal{Y}$ implies that $x \in \mathcal{D}$ and $F(x) = y$.*

**Theorem 2.1.9.** *A subspace of a normed linear space is closed if and only if it is weakly sequentially closed.*

### 2.1.3 Total Variation and Bounded Variation

In this section I will introduce the concept of total variation, functions of bounded variations and some of their properties. Following from Aubert and Kornprobst [AK06], let $\Omega$ be a bounded open set of $\mathbb{R}^N$, and let $u$ be a function in $L^1(\Omega)$. We set

$$\int_\Omega |Du| = \sup \Big\{ \int_\Omega u \; div\varphi dx : \; \varphi = (\varphi_1, \varphi_2, ...\varphi_N) \in \big(\mathcal{C}_0^1(\Omega)\big)^N, |\varphi|_{L^\infty(\Omega)} \leq 1 \Big\} \quad (2.1.1)$$

where $div\varphi = \sum_{i=1}^{N} \frac{\partial \varphi_i}{\partial x_i}(x)$, $dx$ is the Lebesgue measure, $\left(\mathcal{C}_0^1(\Omega)\right)^N$ is the space of continuously differentiable functions with compact support in $\Omega$, $Du$ is the distributional derivative of $u$, and

$$|\varphi|_{L^\infty(\Omega)} = \sup_x \sqrt{\sum_i \varphi_i^2(x)}.$$

This leads to the definition of the space of functions of bounded variations:

**Definition 2.1.10.** *(Definition of Total Variation) For a function $u$ in $L^1(\Omega)$, the total variation of $u$, denoted by $TV[u]$, is defined as*

$$TV[u] = \int_\Omega |Du|.$$

*(Definition of $BV(\Omega)$) The space of functions of bounded variations $BV(\Omega)$ is defined as*

$$BV(\Omega) = \left\{u \in L^1(\Omega), \int_\Omega |Du| < +\infty\right\}$$

In the following part of this dissertation, I will use the notation of $TV[u]$ and $\int_\Omega |Du|$ to denote the total variation of function $u$ indifferently.

**Remark 2.1.11.** *(i) The space $BV(\Omega)$ endowed with the norm*

$$||u||_{BV(\Omega)} = ||u||_{L^1(\Omega)} + \int_\Omega |Du|$$

*is a Banach space [AFP00];*
*(ii) As the distributional derivative of $u$, $Du$ may be decomposed into an absolute continuous part $D^a u$ and a singular part $D^s u$, i.e., $Du = D^a u + D^s u$. In the cases where $u \in W^{1,1}(\Omega)$, the singular part $D^s u$ vanishes, and $Du = D^a u$. This implies*

$$TV[u] = \int_\Omega |Du| = \int_\Omega |\nabla u| dx, \tag{2.1.2}$$

*and* $W^{1,1}(\Omega) \subset BV(\Omega)$.

## 2.2 Regularisation Theory

### 2.2.1 The Moore-Penrose Generalised Inverse

The Moore-Penrose generalised inverse is defined for bounded linear operators on Hilbert spaces. To introduce this concept, I start from the following closely related definitions in the setting of linear inverse problem (1.2.1):

**Definition 2.2.1.** *Let* $A : \mathcal{X} \to \mathcal{Y}$ *be a bounded linear operator between two Hilbert spaces* $\mathcal{X}$ *and* $\mathcal{Y}$*:*
*(i)* $x \in \mathcal{X}$ *is called a least-squares solution of* $Ax = y$ *if*

$$||Ax - y|| = \inf\{||Az - y|| \big| z \in \mathcal{X}\} \tag{2.2.1}$$

*(ii)* $x \in \mathcal{X}$ *is called a best-approximate solution of* $Ax = y$ *if* $x$ *is a least-squares solution of* $Ax = y$*, and*

$$||x|| = \inf\{||z|| \big| z \text{ is least-squares solution of } Ax = y\} \tag{2.2.2}$$

*holds.*

It is important to note that a least square solution may not exist if $\mathcal{R}(A)$ is not closed, which can only happen when this space is infinite. When a least square solution exists, it can be shown that the best-approximate solution denoted above is unique [EHN96]. In many applications, (2.2.2) is not appropriate due to the nature of the solution, and is often replaced by the minimisation of $||Lx||$, where $L$ is usually a differential operator. As one of the most important concepts in inverse problems theory, the Moore-Penrose generalised inverse of $A$ maps an observation $y$ onto the best-approximate solution of $Ax = y$. To see this, firstly we define an operator $\hat{A}$:

**Definition 2.2.2.**

$$\hat{A} := A\big|_{\mathcal{N}(A)^\perp} : \mathcal{N}(A)^\perp \to \mathcal{R}(A) \tag{2.2.3}$$

$\hat{A}$ is well-defined: as $\mathcal{N}(\hat{A}) = \{0\}$, $\hat{A}^{-1}$ exists. The Moore-Penrose generalised inverse hence may be defined as the unique linear extension of $\hat{A}^{-1}$ as follows:

**Definition 2.2.3.** *The Moore-Penrose generalised inverse $A^\dagger$ of $A \in \mathcal{L}(\mathcal{X}, \mathcal{Y})$ is defined as the unique linear extension of $\hat{A}^{-1}$ to*

$$\mathcal{D}(A^\dagger) := \mathcal{R}(A) + \mathcal{R}(A)^\perp \tag{2.2.4}$$

*with*

$$\mathcal{N}(A^\dagger) = \mathcal{R}(A)^\perp, \tag{2.2.5}$$

This definition ensures the existence of the solution $x^\dagger = A^\dagger y$ by including $\mathcal{R}(A)^\perp$ into the domain of $A^\dagger$, while the uniqueness of the solution follows from the injectivity of $\hat{A}$. In other words, for any $y \in \mathcal{D}(A^\dagger)$, it has a unique representation $y = y_1 + y_2$, with $y_1 \in \mathcal{R}(A)$, $y_2 \in \mathcal{R}(A)^\perp$, and $A^\dagger y = \hat{A}^{-1} y_1$.

Further properties of the Moore-Penrose generalised inverse $A^\dagger$ are presented as follows [EHN96],[Nas76],[Gro77]:

**Proposition 2.2.4.** *Let $P$ and $Q$ be the orthogonal projectors onto $\mathcal{N}(A)$ and $\overline{\mathcal{R}(\mathcal{A})}$ respectively, where $\overline{\mathcal{R}(\mathcal{A})}$ is the closure of $\mathcal{R}(A)$. Then $\mathcal{R}(\hat{A}) = \mathcal{N}(A)^\perp$, and the four "Moore-Penrose equations" hold:*

$$\begin{aligned} AA^\dagger A &= A \\ A^\dagger A A^\dagger &= A^\dagger \\ A^\dagger A &= I - P \\ AA^\dagger &= Q\big|_{\mathcal{D}(A^\dagger)} \end{aligned}$$

**Proposition 2.2.5.** *The Moore-Penrose generalised inverse $A^\dagger$ has a closed graph $gr(\hat{A})$. Furthermore, $\hat{A}$ is bounded (i.e. continuous) if and only if $\mathcal{R}(A)$ is closed.*

These properties lead to the two important theorems below regarding the existence and uniqueness of the best-approximate solution:

**Theorem 2.2.6.** *Let $y \in \mathcal{D}(A^\dagger)$. Then $x \in \mathcal{X}$ is a least-squares solution of $Ax = y$ if and only if the normal equation*

$$A^*Ax = A^*y \tag{2.2.6}$$

*holds.*

**Theorem 2.2.7.** *Let $y \in \mathcal{D}(A^\dagger)$. Then $Ax = y$ has a unique best-approximate solution, which is given by*

$$x^\dagger := A^\dagger y.$$

*The set of all least-squares solutions is $x^\dagger + \mathcal{N}(A)$.*

### 2.2.2 Regularisation Method

We use the formal definition of a regularisation method as in Engl, Hanke and Neubauer's book "Regularisation of Inverse Problems"(1997) [EHN96]. In this definition, we seek a stable approximation to the best-approximation $x^\dagger$ in the case when the "exact data" $y$ is not known precisely, but only "noisy data" $y^\epsilon$ with

$$||y^\epsilon - y|| < \epsilon. \tag{2.2.7}$$

is given, where $\epsilon$ is finite and called the "noise level".

**Definition 2.2.8.** *Let $A: \mathcal{X} \to \mathcal{Y}$ be a bounded linear operator between the two Hilbert spaces $\mathcal{X}$ and $\mathcal{Y}$, and $\alpha_0 \in (0, +\infty)$. For every $\alpha \in (0, \alpha_0)$, let*

$$R_\alpha: \mathcal{Y} \to \mathcal{X}$$

*be a continuous (not necessarily linear) operator. The family* $\{R_\alpha\}$ *is called a regularisation or a regularisation operator for the Moore-Penrose generalised inverse* $A^\dagger$*, if for all* $y \in \mathcal{D}(A^\dagger)$*, there exists a parameter choice rule* $\alpha = \alpha(\epsilon, y^\epsilon)$*, such that*

$$\limsup_{\epsilon \to 0} \{ ||R_{\alpha(\epsilon, y^\epsilon)} y^\epsilon - A^\dagger y|| \big| y^\epsilon \in \mathcal{Y}, ||y^\epsilon - y|| \leq \epsilon \} = 0 \tag{2.2.8}$$

*holds. Here,*

$$\alpha : \mathbb{R} \times \mathcal{Y} \to (0, \alpha_0) \tag{2.2.9}$$

*is such that*

$$\limsup_{\epsilon \to 0} \{ \alpha(\epsilon, y^\epsilon) \big| y^\epsilon \in \mathcal{Y}, ||y^\epsilon - y|| \leq \epsilon \} = 0. \tag{2.2.10}$$

*For a specific* $y \in \mathcal{D}(A^\dagger)$*, a pair* $(R_\alpha, \alpha)$ *is called a (convergent) regularisation method for solving* $Ax = y^\epsilon$*, if (2.2.9) and (2.2.10) hold.*

Note that although this definition is given for bounded linear operators, it can be extended to nonlinear problems. In future sections we will recall this definition, and discuss its application in nonlinear cases.

### 2.2.3 Tikhonov Regularisation

To understand the theory of regularisation better, we illustrate it with the classical Tikhonov (or Tikhonov-Phillips) regularisation. In this section we continue with the previous example of linear inverse problem (1.2.1), and assume only the knowledge of linear operator $A$, noisy data $y^\epsilon$ with $||y^\epsilon - y|| < \epsilon$, and noise level $\epsilon$. Our aim is to find an approximated solution for $x^{exact}$ which satisfies

$$Ax^{exact} = y,$$

where $y$ is the unknown exact right hand side. The application of Tikhonov regularisation to problem (1.2.1) corresponds to the solution of the following minimisation for the objective cost functional $J_\alpha(x)$

$$\min_{x\in\mathcal{X}} J_\alpha(x) =: \min_{x\in\mathcal{X}}\{||Ax - y^\epsilon||^2 + \alpha||x||^2\}. \tag{2.2.11}$$

It is easy to check that $J_\alpha(x)$ is quadratic and convex in $x$, hence by taking Fréchet derivative in terms of $x$, its stationary solution $x_\alpha$ minimises $J_\alpha(x)$, and satisfies the equation

$$A^*Ax_{\alpha(\epsilon)} + \alpha x_{\alpha(\epsilon)} = A^*y^\epsilon, \tag{2.2.12}$$

which may be solved as

$$x_{\alpha(\epsilon)} = R_{\alpha(\epsilon)}y^\epsilon, \tag{2.2.13}$$

with the operator $R_{\alpha(\epsilon)}$ defined as

$$R_{\alpha(\epsilon)} := (\alpha I + A^*A)^{-1}A^* : \mathcal{Y} \to \mathcal{X}. \tag{2.2.14}$$

The solution in (2.2.13) is referred to as the **Tikhonov regularisation solution**. It is worth mentioning that in the case of $\epsilon \to 0$ and $\alpha(\epsilon) \to 0$, this solution tends to the best approximated solution of the linear inverse problem (1.2.1). To investigate its error with respect to the true solution $x^{exact}$, consider

$$x_{\alpha(\epsilon)} - x^{exact} = R_\alpha y^\epsilon - R_\alpha y + R_\alpha Ax^{exact} - x^{exact},$$

and use the triangle inequality to write

$$\begin{aligned} ||x_{\alpha(\epsilon)} - x^{exact}|| &\leq ||R_{\alpha(\epsilon)}y^\epsilon - R_{\alpha(\epsilon)}y|| + ||R_{\alpha(\epsilon)}y - x^{exact}|| \\ &\leq \epsilon||R_{\alpha(\epsilon)}|| + ||R_{\alpha(\epsilon)}Ax^{exact} - x^{exact}||. \end{aligned} \tag{2.2.15}$$

The first and second terms in (2.2.15) come respectively from the noise $\epsilon$ in the measurement and from the error due to the approximation $R_\alpha$. We will show later using the tool of singular value decomposition (SVD) that $||R_{\alpha(\epsilon)}||$ is indeed bounded for any regularisation parameter $\alpha > 0$. This term ensures "stability" (not existence) of the inverse of $\alpha I + A^*A$, as a lower bound to its eigenvalues is guaranteed by Gershgorin's Theorem. We may conclude the following from error bound (2.2.15):

- As $\alpha \to 0$, $R_\alpha = (\alpha I + A^*A)^{-1}A^* \to A^\dagger$. This means that the second term of the last inequality in (2.2.15) vanishes, as $||R_\alpha A x^{exact} - x^{exact}|| \to 0$. However, with $A$ being a bounded linear operator, $||A^\dagger||$ is unbounded, which means $||R_\alpha|| \to +\infty$, and the first term in (2.2.15) diverges. This corresponds to the under-regularised case, where the regularisation strength is weak, and the instability in the operator $A^*A$ is not sufficiently removed;

- As $\alpha \to +\infty$, $R_\alpha = (\alpha I + A^*A)^{-1}A^*$ approximates $A^\dagger$ badly, which leads to large values of the second term $||R_\alpha A x - x||$. This corresponds to over-regularisation, where the objective cost functional $J_\alpha(x)$ in (2.2.11) is dominated by the regulariser $\alpha||x||$, and changes the nature of the original inverse problem (1.2.1).

This effect may be illustrated in the Fig (2.1) below (image courtesy: Kirsch (1996)[Kir96]): The two curves represent the two branches in the error bound (2.2.15) against the regularisation parameter $\alpha$ respectively.The third curve on the top is the sum of these two curves, and the best choice $\alpha^*$ is taken to be the minimiser of this sum.

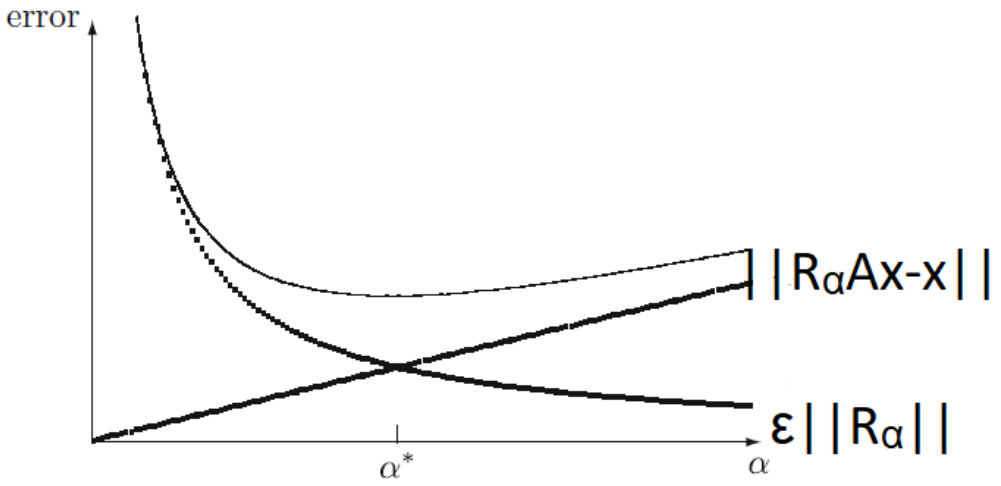



Figure 2.1: Reconstructed solutions with different parameter choices of $\alpha$

### 2.2.4 Singular Value Decomposition

To further understand what regularisation does, we hereby introduce the powerful tool of singular value decomposition (SVD), which is commonly used for linear inverse problems. Here we continue with linear problem (1.2.1) , and seek a solution of the normal equation (2.2.6) for the operator $A$:

**Definition 2.2.9.** *In linear algebra, any $k \times n$ matrix $A$ may be decomposed as*

$$A = U\Sigma V^*, \tag{2.2.16}$$

*where $U$ and $V$ are $k \times k$ and $n \times n$ unitary matrices respectively, and $\Sigma$ is a diagonal $k \times n$ matrix. Then (2.2.16) is called the* ***singular value decomposition*** *of $A$.*

To construct SVD for $A$, note that both $AA^*$ and $A^*A$ are Hermitian matrices, therefore their eigenvalues are real and non-negative, and the corresponding eigenvectors form orthonormal basis, which we denote by $\{u_i\}_{i=1,2,...k}$ and $\{v_i\}_{i=1,2,...n}$ and are the columns of the unitary matrices $U$ and $V$ respectively. Denoting by $\sigma_i^2$ $(i \in \mathbb{N})$ the eigenvalues of the operator $AA^*$ and $A^*A$, the diagonal matrix $\Sigma$ is given by

$\Sigma = diag\{\sigma_1, \sigma_2, ..., \sigma_{rank(A)}, 0, ...0\}$. We have

$$\begin{aligned} Av_i &= \sigma_i u_i, \ i = 1, 2, ...n \\ A^* u_i &= \sigma_i^* v_i, \ i = 1, 2, ...k \\ AA^* u_i &= \sigma_i^2 u_i, \ i = 1, 2, ...n \\ A^* A v_i &= \sigma_i^2 v_i, \ i = 1, 2, ...k \\ \Sigma &= diag\{\sigma_1, \sigma_2, ...\sigma_{rank(A)}, 0, ...0\}, \\ Ax &= \sum_{i=1}^{n} \sigma_i \langle x, v_i \rangle u_i, \quad \forall x \in \mathcal{X}, \\ A^* y &= \sum_{i=1}^{k} \sigma_i \langle y, u_i \rangle v_i, \quad \forall y \in \mathcal{Y}. \end{aligned} \tag{2.2.17}$$

When $A$ is a compact operator on an infinite Hilbert space, then a similar decomposition applies, with a set of equations similar to eq (2.2.17). This relies on a spectral theorem for compact self-adjoint operators [HN01]. In this case $\{\sigma_i^2\}_{i\in\mathbb{N}}$ are the eigenvalues of the self-adjoint operators $AA^*$ and $A^*A$, which accumulates at 0, i.e.

$$\lim_{i\to\infty} \sigma_i = 0. \tag{2.2.18}$$

$\{u_i\}_{i\in\mathbb{N}}$ is the complete set of eigenvectors of $AA^*$ (which forms an orthonormal basis spanning $\overline{\mathcal{R}(A)} = \overline{\mathcal{R}(AA^*)}$), and $\{v_i\}_{i\in\mathbb{N}}$ is the complete set of eigenvectors of $A^*A$ (which forms an orthonormal basis spanning $\overline{\mathcal{R}(A^*)} = \overline{\mathcal{R}(A^*A)}$). The set $\{\sigma_i, u_i, v_i\}$ is called a **singular value system** for $A$.

In either the finite or infinite case, the Moore-Penrose generalised inverse $x^\dagger$, defined in Theorem 2.2.7 for data $y \in \mathcal{D}(A^\dagger)$, can be written as

$$x^\dagger = A^\dagger y = \sum_{i=1}^{\infty} \frac{\langle y, u_i \rangle}{\sigma_i} v_i, \tag{2.2.19}$$

where the sum should be interpreted as a finite sum in the finite case. For the infinite dimensional case, this representation only makes sense if the sum converges, which requires that the data $y$ satisfies the **Picard Criterion**:

$$\sum_{i=1}^{\infty} \frac{|\langle y, u_i \rangle|^2}{\sigma_i^2} < \infty.$$

It can be shown that the Picard Criterion is equivalent to $y \in \mathcal{D}(A^\dagger)$[EK05]. The representation 2.2.19 shows clearly the difference between the finite and the infinite dimensional case, and why regularisation is needed for the latter. For finite dimensional problems with $dim(\mathcal{R}(A)) < \infty$, where the sum in 2.2.19 is finite, any error in the observed data $y$ can only be amplified by a finite scale, which means the third type of ill-posedness (*i.e.* instability) does not occur, though finite dimensional problems can be ill-conditioned, and cause instability in numerical calculations, if the ratio $C$ of the largest to the smallest singular eigenvalue is such that $\log(C) \gtrsim p$, where $p$ is the numerical precision of the matrix entries. For infinite dimensional problems, since the singular eigenvalues satisfy (2.2.18) due to the compactness of operator $A$, we can now see from (2.2.19) why the problem is ill-posed and needs regularisation, since any error in the observed data $y$ is magnified by a factor $\frac{1}{\sigma_i}$ which tends to $\infty$. To stabilise the problem, one approach is to apply the truncated singular value decomposition (**truncated SVD**) [Amm08]. A threshold $\alpha > 0$ is chosen such that all eigenvalues with $\sigma_i < \alpha$ is replaced by 0. This gives rise to the regularised solution

$$x_\alpha = \sum_{\sigma_i \geq \alpha} \frac{\langle y, u_i \rangle}{\sigma_i} v_i, \tag{2.2.20}$$

or

$$x_{\alpha(\epsilon)} = \sum_{\sigma_i \geq \alpha(\epsilon)} \frac{\langle y^\epsilon, u_i \rangle}{\sigma_i} v_i,$$

for noisy observation $y^\epsilon$, where all but the eigenvectors corresponds to eigenvalues with $0 < \sigma_i < \alpha$ are discarded. Note that the threshold $\alpha$ in this case plays the role of the regularisation parameter. Other regularisation methods can be used to stabilise (2.2.19), which are similar to the truncated SVD method. To see how these methods work, we note that truncated SVD can be seen as a modification of the factor $\frac{1}{\sigma_j}$ (the source of ill-posedness!) by a function $g_\alpha(\sigma_i)$ as follows:

$$x_\alpha = \sum_{\sigma_i \geq \alpha} \frac{< y, u_i >}{\sigma_i} v_i = \sum_{i=1}^{\infty} g_\alpha(\sigma_i) < y, u_i > v_i \ ,$$

$$g_\alpha(\sigma_i) = \begin{cases} \frac{1}{\sigma_i} & \text{when} \ \ \sigma_i \geq \alpha \ , \\ 0 & \text{when} \ \ \sigma_i < \alpha \ . \end{cases} \tag{2.2.21}$$

Other regularisation methods can be constructed by simply replacing the amplifying factor $\frac{1}{\sigma_i}$ with a function $g_\alpha(\sigma_i)$ [EK05] such that

$$g_\alpha(\sigma_i) \to \frac{1}{\sigma_i}, \quad as \ \ \alpha \to 0. \tag{2.2.22}$$

In the case of Tikhonov regularisation, it is easy to check that

$$x_\alpha = (A^*A + \alpha I)^{-1} A^* y = \sum_{i=1}^{\infty} \frac{1}{\sigma_i^2 + \alpha} \sigma_i \big\langle y, v_i \big\rangle v_i,$$

which corresponds to the function

$$g_\alpha(\sigma_i) = \frac{\sigma_i}{\sigma_i^2 + \alpha}$$

satisfying (2.2.22).

SVD is a very useful and widely applied tool in linear inverse problems. However,

how to construct a SVD system for a nonlinear operator is not a clearly defined concept. One of the difficulties is the construction of the adjoint operator for a nonlinear operator, which, as noted in Section 1.4 of this dissertation, is not always well-defined. Therefore, when SVD is applied to nonlinear problems, linearisation is usually required first.

### 2.2.5 Choice Rule for Regularisation Parameter

In this section I will discuss different choice rules for the regularisation parameter selection. According to the dependence of the regularisation parameter $\alpha$, they may be characterised into three main categories as described in the table below [EHN96]:

| category | $\alpha$ dependence | example |
|---|---|---|
| a-priori | $\alpha = \alpha(\epsilon)$ | |
| a-posteriori | $\alpha = \alpha(\epsilon, y^\epsilon)$ | discrepancy principle |
| heuristic | $\alpha = \alpha(y^\epsilon)$ | L-curve method, GCV |

Table 2.1: Different categories of the regularisation parameter choice rules according to $\alpha$ dependence

A-priori parameter choice rules depend only on the noise level $\epsilon$ but not the actual (noisy) data $y^\epsilon$, therefore they are not derived from the calculation which involves the residual $||Ax_\alpha^\epsilon - y^\epsilon||$. Usually they are developed before the calculation, and hence named "a-priori". Most of the choice rules implemented in practice are a-posteriori rules or heuristic choice rules of some sorts, *i.e.* $\alpha = \alpha(\epsilon, y^\epsilon)$ or $\alpha = \alpha(y^\epsilon)$.

One of the most well known a-posteriori choice rule is the **Discrepancy Principle** by Morozov [Mor66]. To introduce this method, I continue to work in the setting of the linear inverse problems where only the noisy data $y^\epsilon$ is available. As we have discussed in Section 2.2.3, too weak a regularisation would cause under-fitting, and fail to eliminate instability sufficiently, so we need $\alpha$ large enough to ensure stability. Since we have $||y - y^\epsilon|| \leq \epsilon$, it does not make sense asking for an approximate solution $x_{\alpha(\epsilon)}$ with a residual (or discrepancy) $||Ax_\alpha^\epsilon - y^\epsilon|| < \epsilon$. Therefore, we would like to choose the regularisation parameter $\alpha(\epsilon)$ as the largest possible value such that the

discrepancy is of the same order of the noise level $\epsilon$. This motivates the following definition:

**Definition 2.2.10.** *The regularisation parameter defined via the* ***discrepancy principle*** *is*

$$\alpha(\epsilon, y^\epsilon) := \sup\{\alpha > 0 \big| ||Ax_\alpha^\epsilon - y^\epsilon|| \leq c\epsilon\}, \tag{2.2.23}$$

where $c > 1$ is some positive constant to determine, following the previous discussion of the motivation. More details of how to determine $c$ may be found in [EHN96], [Vai82].

Heuristic choice rules are widely applied in practice, and motivates new developments in adaptive regularisation methods. However, the following theorem by Bakushinskii [Bak84] shows that no heuristic choice rule may be a convergent regularisation method in the sense of Definition 2.2.8 [EHN96]:

**Theorem 2.2.11.** *Let $A : \mathcal{X} \to \mathcal{Y}$ be a bounded linear operator and assume that there is a regularisation $\{R_\alpha\}$ for the Moore-Penrose generalised inverse $A^\dagger$ with a parameter choice rule $\alpha$ which only depends on the noisy observation $y^\epsilon$ but not the noise level $\epsilon$ such that the regularisation method $(R_\alpha, \alpha)$ is convergent for every $y \in \mathcal{D}(A^\dagger)$. Then $A^\dagger$ is bounded.*

This theorem implies that heuristic choice rules work well (or can work well) for ill-conditioned but not for ill-posed inverse problems. Therefore, parameter choice rules of this kind are popular for solving finite dimensional or discrete inverse problems.

One of such choice rules is the so-called **L-curve method** introduced by Hansen in 1992 [Han92]. This method is implemented by plotting the norm of the approximated solution $||x_{\alpha(\epsilon)}||$ against the norm of the residual $||Ax_{\alpha(\epsilon)} - y^\epsilon||$ in a log-log scale over a wide range of $\alpha$. The motivation behind this method is to achieve the balance between these two terms, in the hope of minimising the error between the approximate and exact solution (2.2.15). Analysis of the behaviour of the plot just

described provides heuristic arguments in support of the L-curve method. As large values of regularisation parameter $\alpha$ would lead to over-regularisation, the objective cost functional in these cases will be predominantly taken over by the regularisation term. This means the approximated solution $x_{\alpha(\epsilon)}$ would tend to 0 gradually. Hence the curve will be flat and decay gently in this area. On the other hand, small choices of regularisation parameter $\alpha$ result in instability of the approximated solution, thus blow up its norm $||x_{\alpha(\epsilon)}||$. This causes a steep drop of the curve with a sharp turning. As a consequence, the curve is "L" shaped, with the convex corner representing the balance between over and under regularisation.

A practical challenge in implementing this L-curved method is to locate the sharp, convex corner at the turning. There are usually a wide range of regularisation parameters gathering around this corner, and the optimal one cannot be easily distinguished. In trial-and-error methods the choice of the regularisation parameter using the L-curve is made by visualisation, but in some cases it is possible to use a numerical optimisation strategy. An algorithm providing such optimisation strategy was advocated by Hansen and O'Leary [HO93] who took a geometrical approach to maximise the curvature of the L-curve, *i.e.*

$$\max_{\alpha} \kappa(\alpha) = \frac{\xi''(\alpha)\nu'(\alpha) - \xi'(\alpha)\nu''(\alpha)}{\left(\xi'(\alpha)^2 + \nu'(\alpha)^2\right)^{\frac{3}{2}}}, \tag{2.2.24}$$
$$\xi(\alpha) = \log||y^{\epsilon} - Ax_{\alpha(\epsilon)}||,\ \nu(\alpha) = \log||x_{\alpha(\epsilon)}||,$$

and the optimal value by the L-curve method is chosen to be the $\alpha$ which maximises (2.2.24).

Even though this method works well for non-smooth curves with a sharp corner, it will be harder to select the optimal $\alpha$ based on this rule for curves which are generically more smooth and do not possess a dramatic turning. To remedy this drawback, Regińska [Reg96] has suggested an alternative definition, which defines the "corner" point of an L-curve in the following sense:

**Definition 2.2.12.** *A point $C = (\xi(\alpha_*), \nu(\alpha_*))$ is the corner of the L-curve $\mathcal{L}$ if $\mathcal{L}$ is concave in a neighbourhood of $C$, and the tangent of $\mathcal{L}$ at $C$ has slope -1.*

This definition leads to a new interpretation of the L-curve method, as well as the following proposition:

**Proposition 2.2.13.** *The point $C = (\xi(\alpha_*), \nu(\alpha_*))$ is a corner of the L-curve in the sense of Definition 2.2.12 if and only if the function*

$$\psi(\alpha) := ||y^\epsilon - Ax_{\alpha(\epsilon)}||||x_{\alpha(\epsilon)}|| \tag{2.2.25}$$

*has a local minimum at $\alpha = \alpha_*$.*

The detail of the proof may be found in [EHN96].

Another well-known heuristic choice rule is the **Generalised Cross-Validation** (GCV). This model was originally introduced by Wahba (1979) [GHW79] in the context of the linear regression, and is essentially Tikhonov regularisation applied to statistics. It applies to linear and finite dimensional problems, and depends on the assumption that the data perturbation $y^\epsilon - y$ is discrete white noise, i.e.

$$\begin{aligned} \mathbb{E}[y^\epsilon - y] &= 0, \\ \mathbb{E}[(y^\epsilon - y)(y^\epsilon - y)^T] &= \sigma^2 I, \end{aligned}$$

where $\mathbb{E}[]$ denotes expectation, and $\sigma^2$ is the variance of the noise. Note that the aim for GCV model is to find a suitable estimate $\hat{x}$ (which is called a Ridge estimator in this context) satisfying the constraint

$$||x|| = \gamma$$

such that the residue $||y^\epsilon - Ax||$ is minimised with the presence of noise $\epsilon$. By introducing the Lagrangian, we can see this is equivalent to the minimisation of the

objective cost functional (2.2.11) in Tikhonov regularisation. Methods for the determination of $\gamma$ and $\alpha$ in this context may be found in [Gol73].

Just as the previously discussed L-curve method, the major merit of GCV model is that it does not need knowledge of the noise level $\epsilon$ characterised by the standard deviation of its distribution, which is usually not known in practice. Historically, the GCV estimate has been successfully implemented by Wahba, World and Craven in curve smoothing, numerical differentiation, optimal smoothing of density and spectral density estimates, and demonstrated to be effective [CW79],[Wah77],[WW75b],[WW75a].

### 2.2.6 Choice of the Fidelity Term

So far we have only seen fidelity term in the form of $\mathcal{L}^2$ norm in the previous example of Tikhonov regularisation. In practice, the choice of fidelity usually depends on the statistical property of the noise in the observed noisy data $y^\epsilon$ [Stu10], [Idi13]. It is in general vital to choose an appropriate form of functional for the data fidelity term, as a bad choice would result in either inaccurate restoration or slow convergence of the algorithm applied. Calatroni *et al* [CDlRS17] have shown the derivation of different types of fidelity terms by assuming different noise distributions using the power of the MAP estimator. A brief summary of different choices of fidelity terms against noise distribution is presented below:

| Type of noise | Fidelity term |
|---|---|
| Gaussian | $\|\|(Ax - y^\epsilon)\|\|_2^2$ |
| Impulse (salt and pepper) | $\|\|Ax - y^\epsilon\|\|_1$ |
| Uniform | $\|\|Ax - y^\epsilon\|\|_\infty$ |

Table 2.2: Appropriate choices of data fidelity terms for different noise distributions in objective cost functional $J_\alpha(x)$ of (2.2.11)

Here $||\cdot||_2$, $||\cdot||_1$ and $||\cdot||_\infty$ refer to the 2-norm, 1-norm and infinity norm respectively.In the cases where the noise distribution is mixed (*i.e.* multiple types of

noise corrupt the observation at the same time), a combination of the corresponding fidelities often applies [CDlRS14], [DlR14], [LH13].

## 2.3 Numerical Algorithms

In this section we consider the minimisation problem

$$\min_{x\in\mathbb{R}^n} f(x) \tag{2.3.1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is differentiable, and discuss some search algorithms applied in later chapters of this dissertation.

### 2.3.1 Line Search Method

According to Sun and Yuan [SY06], line search, also called one-dimensional search, refers to an optimization procedure for univariable functions. It is the base of multivariable optimization, with the method being defined by the iterative scheme

$$x_{k+1} = x_k + \alpha_k d_k \tag{2.3.2}$$

for each given $x_k$. The key of the algorithm is to find a search direction $d_k$ and a step size $\alpha_k$ to update $x_{k+1}$. For each given search direction $d_k$, we define a function in $\alpha$ as

$$\phi(\alpha) = f(x_k + \alpha d_k),$$

if step size $\alpha_k$ is chosen in such a way that the objective function $f(x)$ in direction $d_k$ is minimised, *i.e.*

$$f(x_k + \alpha_k d_k) = \min_{\alpha} f(x_k + \alpha d_k),$$

which is equivalently

$$\phi(\alpha_k) = \min_{\alpha} \phi(\alpha),$$

then such a line search method is called an exact line search, or optimal line search, and $\alpha_k$ is called the optimal step size. In the cases where the objective function has acceptable descent amount, *i.e.*

$$f(x_k + \alpha_k d_k) < f(x_k), \tag{2.3.3}$$

which is equivalently

$$\phi(\alpha_k) < \phi(0),$$

such a line search is called an inexact (or approximate/accepted) line search. In Chapter 5.2 of this dissertation, the modified alternating minimisation algorithm (Algorithm 5.2.1) employs an inexact line search in the inner loop minimisation for the image intensity $u$ in the multiplicative models.

### 2.3.2 Steepest Descent Method

With the assumption of the function $f$ being differentiable, linearisation of (2.3.3) leads to

$$f(x_k + \alpha_k d_k) \approx f(x_k) + \alpha_k \big\langle \nabla f(x_k), d_k \big\rangle.$$

Ideally we would like to choose the normalised search direction $d_k$ such that

$$d_k = \arg\min_{||d||=1} \big\langle \nabla f(x_k), d \big\rangle,$$

in other words, the direction where the function $f$ decreases the fastest. It may be shown that

$$d_k = -\nabla f(x_k)$$

satisfies this criterion, which is known as the **steepest descent** search method.

### 2.3.3 Conjugate Gradient Method

In the minimisation of non-quadratic functions $f$, which arise when seeking approximate solutions to nonlinear problems, a line search method known as the conjugate gradient method is usually applied [DHL+99]. Again it follows the scheme of line search (2.3.2), with search direction $d_k$ defined as

$$d_{k+1} = \begin{cases} -g_k & \text{for } k = 1, \\ -g_k + \beta_k d_{k-1} & \text{for } k \geq 2 \end{cases} \tag{2.3.4}$$

where $\beta_k$ is a scalar, and $g_k = \nabla f(x_k)$ is the gradient of the objective functional evaluated at $x_k$. There are various ways to define $\beta_k$, the most well-known are the following Fletcher-Reeves (FR), Polak-Ribière (PR), and Hestenes-Stiefel (HS) formulae

$$\beta_k^{FR} = ||g_k||^2 / ||g_{k-1}||^2, \tag{2.3.5}$$

$$\beta_k^{PR} = g_k^T (g_k - g_{k-1}) / ||g_{k-1}||^2, \tag{2.3.6}$$

$$\beta_k^{HS} = g_k^T (g_k - g_{k-1}) / d_{k-1}^T (g_k - g_{k-1}), \tag{2.3.7}$$

where $||\cdot||$ denotes the $L_2$ norm.

It can be proved that in the case of linear problems when $f$ is quadratic, all three formulas for $\beta_k$ are equivalent. This is because $span\{g_1, g_2, ..., g_k\}$ equals $span\{d_1, d_2, ..., d_k\}$, $k \in \mathbf{N}$, as the $k$th Krylov subspace. It can also be shown

that for finite dimensional linear problems with dimension $n$, the convergence is guaranteed within $n$ iterations. However, for nonlinear problems, the expression in (2.3.5),(2.3.6) and (2.3.7) are usually not equivalent, and mathematicians are still investigating which is the best choice in different situation. So far it has been shown that Fletcher-Reeves (2.3.5) is convergent on general functions, while Polak-Ribière (2.3.6) can occasionally lead to infinite cycles without converging. Although Polak-Ribière usually converges much more quickly ([She94]), it is also in general more difficult to calculate the gradient $g_k$ and the step size $\alpha_k$ for nonlinear problems. Sometimes $\alpha_k$ has no closed form expression and hence requires a numerical solution in every iteration, which may take large computational resources.

### 2.3.4 Method of Feasible Directions

According to Zoutendijk [Zou70], a search method falls into the category of the method of feasible directions, if it solves for problem (2.3.1) along the following lines:

1. Starting point $x_0$ satisfies $f(x_0) < \mu$ for some $\mu$, *i.e.* set $\left\{x \in \mathbb{R}^n \middle| f(x) \leq \mu\right\}$ is bounded and non-empty for some $\mu$,
2. Suppose points $x_m, m \leq k$ have already been determined. Then
a. find a search direction $d_m$ such that it guarantees $\left\langle \nabla f(x_k), d_k \right\rangle < 0$;
b. find step length $\alpha_k > 0$.
c. set $x_{k+1} = x_k + \alpha_k x_k$.
3. Repeat with $k := k + 1$ until some stopping criterion is met.

According to Zoutendijk (1970) [Zou70], a method of feasible directions is called usable if at least one of the accumulation points of the sequence $\{x_k\}_{k\in\mathbb{R}}$ is a local stationary point of $f(x)$.

## 2.4 Iterative Methods

### 2.4.1 Landweber Iteration

We shall consider here Landweber iteration in the discrete setting, since the new methods developed in later chapters of this dissertation are in the discrete setting. Therefore, we follow Hansen [Han09] to write the basic form of this method for a linear problem $Ax = y$ as:

$$x^{[k+1]} = x^{[k]} + \omega A^*(y - Ax^{[k]}) \tag{2.4.1}$$

where $\omega$ is a real number that must satisfy

$$0 < \omega < 2||A^*A||^{-1} = \frac{2}{\sigma_1^2}, \tag{2.4.2}$$

where $\sigma_1$ is the largest eigenvalue of matrix $\Sigma$ in the singular value decomposition of $A$ (see Section 2.2.4 for details). The purpose of this condition (2.4.2) is to ensure the non-expansion of linear operator $I - \omega A^*A$, which is guaranteed by condition $||A||^2 < \frac{2}{\omega}$ [EHN96]. It is shown by Hansen [Han09] that given the kth iterate as

$$x^{[k]} = V\Phi^{[k]}\Sigma^{-1}U^*y \tag{2.4.3}$$

where $\Phi^{[k]} = diag(\varphi_1^{[k]}, \varphi_2^{[k]}, ..., \varphi_n^{[k]})$ with diagonal elements being the filter factors for $x^{[k]}$ given by

$$\varphi_i^{[k]} = 1 - (1 - \omega\sigma_i^2)^k, \tag{2.4.4}$$

by substituting (2.4.3) and (2.4.4) into the Landweber iteration scheme (2.4.1) using SVD representation of matrix $A$, the (k+1)th iterate $x^{[k+1]}$ also satisfies (2.4.3) and (2.4.4). This demonstrates that (2.4.3) - (2.4.4) are the solution representation of Landweber scheme (2.4.1).

Due to the semi-convergence property of Landweber iteration, a stopping crite-

rion is required in order to choose an appropriate iteration index and prevent the over-regularisation of solution (2.4.3). One way to do this is to set an arbitrary value for $\varphi_i^{[k]}$, which is a function of $\sigma_k^2$ according to (2.4.4). This will lead to a break-point $\sigma_{break}^{[k]}$ in the filter factors $\sigma_i$ (also the eigenvalues of $\Sigma$ in the SVD of $A$). Note this break-point $\sigma_{break}^{[k]}$ plays the similar role as threshold $\gamma$ in the truncated SVD of (2.2.20): it plays the role of a regularisation parameter and excludes eigenvector components of smaller singular values in the solution. Note that for a fixed $\varphi_i^{[k]}$, the value of $\sigma_{break}^{[k]}$ decreases as iteration index $k$ increases. This explains the semi-convergence of the Landweber iteration: as iteration index $k$ increases, the solution tends to the Moore Penrose generalised inverse and leads to over-regularisation.

Hansen [Han09] also investigated the convergence rate of the linear Landweber iteration scheme (2.4.1). He has shown numerically that for each fixed $\varphi_i^{[k]}$, as iteration index $k$ increases, the singular value filter $\sigma_{break}^{[k]}$ reduces at a small rates as parameter $\omega$ varies. This illustrates the slow semi-convergence towards the exact solution, which is a main drawback of this method.

Apart from linear problems, Landweber iterative method is widely applied particularly to nonlinear problems in the form of

$$F(x) = y, \quad F : \mathcal{X} \to \mathcal{Y}.$$

Being an analogy of the linear scheme (2.4.1), the nonlinear Landweber iteration scheme may be formulated as

$$x_{k+1} = x_k + \omega F'(x_k^*)(y - F(x_k), \ k \in \mathbb{N}. \tag{2.4.5}$$

Here we assume differentiability of the operator $F$, and $F'$ is the Fréchet derivative of the operator $F$. It is worth pointing out that in the literature, sometimes the relaxation parameter $\omega$ is omitted from (2.4.1) and (2.4.5) with the introduction of an additional assumption $||A|| < 1$ or $||F'(x)|| < 1$ for all $x$ in some neighbourhood

locally [EHN96], [KS08]. We can interpret this setting by scaling down the operator $A$ and $F'$ accordingly. Without loss of generality, we will proceed in the setting where the relaxation parameter $\omega$ is omitted.

One way to choose the stopping criterion of nonlinear Landweber iteration scheme is, to employ the discrepancy principle [KS08]. It states that the iteration should stop after $k_* = k_*(\epsilon, y^\epsilon)$ steps, with

$$||y^\epsilon - F(x^\epsilon_{k_*})|| \leq c\epsilon \leq ||y^\epsilon - F(x^\epsilon_k)||, 0 \leq k \leq k_*,$$

where $c > 0$ is some constant parameter to be chosen. Further details about how to make an appropriate choice for this parameter may be found in [Sch95]. Extensive studies on convergence and convergence rates of the scheme have been subsequently carried out by Kaltenbacher *et al* (2008) [KS08], where topics like uniform cone condition, modified Landweber iteration and Landweber-Kaczmarz methods are discussed.

## 2.5 Model by Charbonnier *et al*

In this section, we introduce a spatially dependent regularisation model by Charbonnier *et al* [CBFAB97]. This model is concerned with image deblurring problem with the presence of noise in the setting of (1.3.1), and works well particularly for the edge reconstruction of piecewise constant images. The problem is discretised, so the variables in (1.3.1) are $N \times N$ matrices. Without loss of generality, we consider the unit domain $\Omega = [0, 1] * [0, 1]$ with a discretisation $h > 0$ being the mesh size. This gives rise to the matrix representations of image intensities $u_{i,j}$, $u^{exact}_{i,j}$, $f_{i,j}$ and $f^\epsilon_{i,j}$ with $1 \leq i, j \leq N$, $N = \frac{1}{h}$. To solve this problem, the proposed objective functional has the form

$$J(u) = \sum_{(i,j)} [f^\epsilon_{i,j} - (\mathcal{K}u)_{i,j}]^2 + \lambda \{ \sum_{(i,j)} \varphi[(D_x u)_{i,j}] + \sum_{(i,j)} \varphi[(D_y u)_{i,j}] \}. \qquad (2.5.1)$$

where $D_x$ and $D_y$ denote finite difference operators in $x$ and $y$ direction respectively, at each pixel $(i, j)$:

$$(D_x u)_{i,j} = (u_{i,j} - u_{i,j-1})/h;$$
$$(D_y u)_{i,j} = (u_{i,j} - u_{i-1,j})/h;$$

Note that the fidelity is the standard discrete $L^2$ setting, and the noise $\eta$ is assumed to be Gaussian type. $\varphi$ is called the potential function, and should have a set of natural properties. It is responsible for the edge-preservation of the reconstructed image. There have been discussions in the literature on how to choose an appropriate form of this function: Geman and Reynolds [GR92] prefer functions with a finite asymptotic behaviour, while most of the other researchers favour convex functions to guarantee the uniqueness of the solution [BS93], [SS95], [LH89] and [Lan90]. Charbonnier *et al* [CBFAB97] take all these opinions above into account, and impose the following properties to the potential function $\varphi(t)$:

**Basic Assumptions for $\varphi(t)$:**
(a) $\varphi(t) \geq 0,\ \forall t \geq 0$, and $\varphi(0) = 0$;
(b) $\varphi(t) = \varphi(-t)$,
(c) $\varphi(t)$ being continuously differentiable;
(d) $\varphi'(t) \geq 0,\ \forall t \geq 0$;
**Edge-Preserving Assumptions for $\varphi(t)$:**
(e) $\frac{\varphi'(t)}{2t}$ being continuous and strictly decreasing on $t \in (0, +\infty)$;
(f) $\lim_{t\to\infty} \frac{\varphi'(t)}{2t} = 0$;
(g) $\lim_{t\to 0^+} \frac{\varphi'(t)}{2t} = M$;
**Assumptions for Convergence Proof:**
(h) $\varphi'''(0) = 0$;
(i) $\varphi^{(4)}(t)$ exists;

As Nikolova and Chan [NC07] have demonstrated, it may be slow in general to minimise a functional $J(u)$ in the form of (2.5.1) over $u$ directly for piecewise constant

| Function | Expression of $\varphi$ | Weight Func. $\frac{\varphi'(t)}{2t}$ | Dual Func.$\psi(\omega)$ | Ref. |
|---|---|---|---|---|
| $\varphi_{GM}$ | $\frac{t^2}{1+t^2}$ | $\frac{1}{(1+t^2)^2}$ | $\omega - 2\sqrt{\omega} + 1$ | [GM85] |
| $\varphi_{HL}$ | $\log(1+t^2)$ | $\frac{1}{(1+t^2)}$ | $\omega - \log\omega - 1$ | [LH89] |
| $\varphi_{HS}$ | $2\sqrt{1+t^2} - 2$ | $\frac{1}{\sqrt{1+t^2}}$ | $\omega + \frac{1}{\omega} - 2$ | [CABFB94] |

Table 2.3: Different types of edge-preserving potential functions, and their associated weight functions and corresponding dual functions

images $u(\mathbf{x})$, as for most pixels $(i,j)$, the function $\varphi[(D_x u)_{i,j}]$ and $\varphi[(D_y u)_{i,j}]$ will be zero due to the sparsity of the input image, which causes the slow convergence of the gradient method. This motivates us to formulate a different objective cost functional by applying the following theorem from Charbonnier *et al* [CBFAB97] and Geman and Reynolds [GR92]:

**Theorem 2.5.1.** *If $\varphi$ is a function satisfying all the assumptions (a) to (g) above, then*
*(1) There exists a strictly convex and decreasing function $\psi : (0, M] \to (0, \beta]$, where*

$$\beta = \lim_{t \to +\infty} (\varphi(t) - t^2 \frac{\varphi'(t)}{2t}) \tag{2.5.2}$$

*such that*

$$\varphi(t) = \inf_{0 < \omega \leq M} (\omega t^2 + \psi(\omega)) \tag{2.5.3}$$

*(2) For every fixed $t \geq 0$, the value $\omega_t$ for which the minimum is reached, i.e., such that*

$$\inf_{0 < \omega \leq M} (\omega t^2 + \psi(\omega)) = \omega_t t^2 + \psi(\omega_t)$$

*is unique and given by*

$$\omega_t = \frac{\varphi'(t)}{2t} \tag{2.5.4}$$

The proof of this theorem is presented in Appendix A of this dissertation. Now we will apply this theorem to the original cost functional $J(u)$. To do this, we introduce complementary variables $(b_x)_{i,j}$ and $(b_x)_{i,j}$ for all pixels $(i,j)$ which plays the role of $\omega$ in the theorem. It is then possible to obtain another objective cost functional $J^*(u, b_x, b_y)$ which satisfies

$$J(u) = \inf_{b_x, b_y} J^*(u, b_x, b_y).$$

The explicit form of this cost functional may be expressed as follows:

$$J^*(u, b_x, b_y) = \sum_{(i,j)} [f^\epsilon_{i,j} - (\mathcal{K}u)_{i,j}]^2 + \lambda \{ \sum_{(i,j)} \varphi^*[(D_x u)_{i,j}, (b_x)_{i,j}] + \sum_{(i,j)} \varphi^*[(D_y u)_{i,j}, (b_y)_{i,j}] \}. \tag{2.5.5}$$

where $\varphi^*(t, \omega)$ is a function quadratic in $t$ when $\omega$ is fixed, and such that

$$\varphi(t) = \inf_{\omega} \varphi^*(t, \omega). \tag{2.5.6}$$

To calculate $\varphi^*(t, \omega)$, $\psi(\omega)$ and the minimiser $\omega_t$ explicitly, we consider the following remark, which is essentially a summary of the proof of Theorem 2.5.1:

**Remark 2.5.2.** *Theorem 2.5.1 shows that for any given function $\varphi(t)$ satisfying conditions (a) to (g), a dual function $\psi(\omega)$ can be formulated as*

$$\begin{aligned} \theta(t) &= \varphi(\sqrt{t}); \\ \psi(\omega) &= \theta[(\theta')^{-1}(\omega)] - \omega \cdot (\theta')^{-1}(\omega), \end{aligned} \tag{2.5.7}$$

*and*

$$\omega_t := \arg\min_{\omega} \{\omega t^2 + \psi(\omega)\} = \frac{\varphi'(t)}{2t}. \tag{2.5.8}$$

*This leads to the so-called half quadratic minimisation introduced by [CBFAB97], [CABFB94], i.e., for any $\varphi(t)$ satisfying the conditions (e)-(g), it is possible to find*

*a function $\varphi^*(t,\omega)$ which is quadratic in $\omega$ and satisfies (2.5.6). It may be easily spotted from Theorem 2.5.1(1) that*

$$\varphi^*(t,\omega) = \omega t^2 + \psi(\omega). \tag{2.5.9}$$

It is worth pointing out that in this discrete setting, when $(b_x, b_y)$ is chosen and fixed, the new objective cost functional $J^*(u, b_x, b_y)$ takes the form of

$$\begin{aligned} J^*(u,b_x,b_y) &= \sum_{(i,j)}[f^\epsilon_{i,j} - (\mathcal{K}u)_{i,j}]^2 \\ &+ \lambda\{\sum_{(i,j)}(D_x u)^2_{i,j}(b_x)_{i,j} + \psi((b_x)_{i,j}) + \sum_{(i,j)}(D_y u)^2_{i,j}(b_y)_{i,j} + \psi((b_y)_{i,j})\}, \end{aligned} \tag{2.5.10}$$

hence is quadratic in the primal variable $u$. This is known as the "half-quadratic" nature of the objective functional. The following alternating minimisation algorithm is applied by Charbonnier *et al* [CBFAB97],[CABFB94] to minimise $J^*(u, b_x, b_y)$ in both $u$ and auxiliary variables $b_x$, $b_y$.

**Algorithm 2.5.3.** ***Alternating minimisation***
*Step 0: Initialization: choose $u^0 = f^\epsilon$ to start with;*
*Repeat*
*Step 1:* $(b_x^{n+1}, b_x^{n+1}) = \arg\min_{b_x,b_y}[J^*(u_n, b_x, b_y)]$
*Step 2:* $u^{n+1} = \arg\min_u[J^*(u, b_x^{n+1}, b_y^{n+1})]$
*Until convergence.*

This minimisation strategy is the key part of the adaptive regularisation by Charbonnier *et al* [CBFAB97]: It minimises in the primal and dual variables alternatively, hence improving the efficiency of the direct minimisation problem $\min_u J(u)$. To justify this algorithm theoretically, consider

$$\min_u J(u) = \min_u \min_{b_x,b_y} J^*(u,b_x,b_y) = \min_{b_x,b_y}\min_u J^*(u,b_x,b_y), \tag{2.5.11}$$

The second equality comes from the exchange of the minimisation order, which is always possible in the discrete setting of the problem. By introducing

$$G(b_x, b_y) = \min_u J^*(u, b_x, b_y),$$

(2.5.11) becomes

$$\min_u J(u) = \min_b G(b_x, b_y).$$

Recall that $J^*(u, b_x, b_y)$ is quadratic in $u$ for each pair of chosen $b := (b_x, b_y)$, so there exists a unique $u_b$ such that

$$u_b = \arg\min_u J^*(u, b_x, b_y), \ \forall\ b = (b_x, b_y).$$

This leads to the following theorem:

**Theorem 2.5.4.** *if* $\hat{b} = \arg\min_b G(b)$, *then* $u_{\hat{b}} = \arg\min_u J(u)$.

The proof of this theorem is shown in Appendix B. This theorem provides the foundation of the alternative minimisation algorithm in Algorithm 2.5.3. It shows that all the minimisers of $G(b)$ yield solution $u_{\hat{b}}$ which minimises $J(u) = \min_b J^*(u, b)$. According to Theorem 2.5.1, the minimisation in the auxiliary variables $(b_x, b_y)$ in Step 1 has closed form solution

$$(\hat{b}_x)_{i,j} = \frac{\varphi'[(D_x u_{\hat{b}})_{i,j}]}{2(D_x u_{\hat{b}})_{i,j}}, \ (\hat{b}_y)_{i,j} = \frac{\varphi'[(D_y u_{\hat{b}})_{i,j}]}{2(D_y u_{\hat{b}})_{i,j}}, \forall\ i, j$$

while the minimisation in the primal variable $u$ in Step 2 is quadratic, hence its unique minimiser may be obtained via the first order optimality condition. The convergence of this alternative minimisation algorithm may be guaranteed by the following theorem:

**Theorem 2.5.5.** *Let* $\varphi$ *be a potential function satisfies all the assumptions (a)-(i) previously, then*
*(i) The sequence of the objective cost functional* $J_n = J(u_n) = J^*(u_n, b_x^{n+1}, b_y^{n+1})$ *is*

*convergent;*
*(ii) We have* $b_x^{n+1} - b_x^n \to 0$ *and* $b_y^{n+1} - b_y^n \to 0$ *as* $n \to \infty$*;*
*(iii) If the potential function* $\varphi$ *is convex, and the linear operator* $\mathcal{K}$ *in (1.3.1) has full rank, then the image intensity* $u_n \to u_{\hat{b}}$*, i.e. it tends to the unique global minimiser of* $J(u) = \arg\min_{b_x,b_y} J^*(u, b_x, b_y)$.
*(iv) If the potential function* $\varphi$ *is convex, whether* $\mathcal{K}$ *has full rank or not, then* $D_x u_n \to D_x u_{\hat{b}}$*, and* $D_y u_n \to D_y u_{\hat{b}}$.

The detailed proof of this theorem is presented in Appendix C of this dissertation. Note that claim (i) and (ii) will always hold regardless of the properties of the potential function $\varphi$ and the linear operator $\mathcal{K}$, while (iii) and (iv) requires further assumptions such as the convexity of $\varphi$ and the rank of linear operator $\mathcal{K}$.

As Theorem 2.5.5 has shown (see Appendix C of this dissertation for proof), when $\varphi$ is convex, either the image intensity $u_n$ or its contrast $D_x u_n$ and $D_y u_n$ will converge to the global minimiser of the objective cost functional $J(u) = \inf_{b_x,b_y} J^*(u, b_x, b_y)$, *i.e.* $u_{\hat{b}}$ or $D_x u_{\hat{b}}$ and $D_y u_{\hat{b}}$. It is then natural to ponder on the cases with non-convex choices of $\varphi$. Unfortunately, so far there are only numerical results available in these cases, and no conclusion may be drawn theoretically about where Algorithm 2.5.3 would direct its solution to. With the absence of the convexity, a reasonable guess would be some local minimiser or just critical points of the objective cost functional $J(u) = J^*(u, b_x, b_y)$. Despite this analytical ambiguity, the algorithm seems to work well numerically and produces satisfactory regularisation solution regardless of the convexity of the potential function, as Charbonnier *et al* has demonstrated [CBFAB97].

Unlike some other adaptive regularisation methods (e.g. Dong *et al* [DHRC11]) which designates their regularisation parameter as a spatially dependent function and self-adjusts its value pixel-wisely by a local error estimator, this regularisation method by Charbonnier *et al* [CBFAB97] keeps a scalar regularisation parameter $\lambda$ in the objective cost functional $J(u) = J^*(u, b_x, b_y)$, and determines its value by

trial and error. The spatial dependence of the regularisation is introduced through the auxiliary variables $b = (b_x, b_y)$, which acts as a "discontinuity map" and adjusts the regularisation effect pixel-wisely. The self-adaptive effect is brought in by the alternative minimisation algorithm, which balances the image intensity and the auxiliary variables. It is worth noticing that the multiplicative regularisation, which is to be discussed in the rest of the thesis, is a self-adaptive regularisation method with spatial dependence. This method inherits the merit of the Charbonnier model [CBFAB97], and works well particularly in the case of piecewise constant images.

## 2.6 Miscellaneous

**Gershgorin Theorem**

The Gershgorin theorem was first published in 1931, and can be used to bound the size of the eigenvalues of a square matrix [Sho07]. Let $A = (a_{i,j}) \in \mathbb{C}_{n\times n}$ be a $n \times n$ square matrix with complex entries, the $i$th Gershgorin disk $D^i$ is defined as

$$D^i = B\left(a_{i,i}, \sum_{j\neq i, j=1}^{n} |a_{i,j}|\right) = \left\{ z \in \mathbb{C} \middle| |z - a_{i,i}| \leq \sum_{j\neq i, j=1}^{n} |a_{i,j}| \right\}, \quad \forall\ i = 1, 2, ...n,$$

the Gershgorin Theorem states that

**Theorem 2.6.1.** *1. Every Eigenvalue of $A$ lies in some Gershgorin disk $D^i$.*
*2. If $M$ is the union of $m$ disks $D^i$ such that $M$ is disjoint from all other disks of this type, then $M$ contains precisely $m$ Eigenvalues of $A$ (counting multiplicity).*

Generally speaking, this theorem says that if the off-diagonal entries of a square matrix over the complex numbers have small norms, then its eigenvalues are "dominated" by the diagonal elements of the matrix, and are similar to the diagonal entries in norm.

# Chapter 3

# Multiplicative Regularisation

## 3.1 Problem Statement

In this dissertation, we shall work on the imaging problem (1.3.1), which was introduced in Section 1.3 and is specified below, as well as a few other test images. We define the linear operator $\mathcal{K}$ that denotes the blurring kernel by:

$$[\mathcal{K}u](\mathbf{x}) = \int_{\Omega} K(\mathbf{x} - \mathbf{x}')u(\mathbf{x}')d\mathbf{x}',$$

with

$$K(x_1, x_2) = \begin{cases} \frac{1}{\pi\zeta^2} & \text{if } \sqrt{x_1^2 + x_2^2} < \zeta, \\ 0 & \text{elsewhere} \end{cases} \tag{3.1.1}$$

Here $\zeta > 0$ is to be specified in the numerical example. In the exact case we have $\mathcal{K}u^{exact} = f$, where $f$ is the blurred image uncorrupted by noise.
The corrupted, noisy image $f^\epsilon$ in (1.3.1) satisfies

$$||f^\epsilon - f|| \leq \epsilon.$$

Here $\epsilon$ is considered as the noise level. To solve this problem, we want to reconstruct the original image $u$ from the model. We do this later using various multiplicative regularisation models and the Charbonnier model [CBFAB97] presented later in this thesis. We will restrict our problem to the finite dimensional and discrete setting, and consider the discretised $\mathcal{L}^2$ space. Since the problem in the infinite dimensional case is ill-posed, because the operator $\mathcal{K}$ is compact, the matrix that arises after discretisation is ill-conditioned and the discretised system is also very sensitive to noise and requires regularisation.

For convenience we will use $||\cdot||_2$ and $\left\langle \cdot \right\rangle_2$ to denote the $L^2$ norm and inner product of the discrete setting in space $\mathbb{R}^{N\times N}$. For any $u, v \in \mathbb{R}^{N\times N}$, $u = u_{i,j}, v = v_{i,j}, 1 \leq i, j \leq N$,

$$||u||_2 = \frac{1}{N^2}\sum_{(i,j)} u_{i,j}^2,$$
$$\left\langle u, v \right\rangle_2 = \frac{1}{N^2}\sum_{(i,j)} u_{i,j}v_{i,j}.$$

## 3.2 Model Formulation

Multiplicative regularisation refers to a type of regularisation in which the cost functional to be minimised is constructed as the product of the fidelity and a regularisation term, rather than the more usual sum of two such terms. For multiplicative type of regularisation, Abubakar *et al* [AvdBHB04] proposed to solve the following cost functional in an iterative manner:

$$C_n(u) = F(u)F_n^{multi}(u), \tag{3.2.1}$$

where, in the discrete setting,

$$F(u) = \sum_{(i,j)} [(\mathcal{K}u)_{i,j} - f^{\epsilon}_{i,j}]^2$$

is the data fidelity term, and

$$F^{multi}_{n}(u) = \sum_{(i,j)} (b^2_n)_{i,j}[(D_x u)^2_{i,j} + (D_y u)^2_{i,j}] + \sum_{(i,j)} \delta^2_n (b^2_n)_{i,j}$$

is the regularisation term, where $b_n$ is a weight function, and $\delta_n$ is a scalar parameter, both to be determined iteratively as described in Algorithm 3.2.1. The general framework for solving (3.2.1) iteratively is described as follows:

**Algorithm 3.2.1.** ***Multiplicative regularisation algorithm by Abubakar et al***

*Step 0: Initialization:* $u_0 = f^{\epsilon}$*; choose a value for* $\delta_0$*; set* $n = 0$*.*

*Repeat*

*Step 1: Update weight function* $b^2_{n+1}$*:*

$$(b^2_{n+1})_{i,j} = \frac{1}{(D_x u_n)^2_{i,j} + (D_y u_n)^2_{i,j} + \delta^2_n} \ \forall \ i, j = 1, 2, ...N. \tag{3.2.2}$$

*Step 2: Update steering parameter* $\delta^2_{n+1}$*:*

$$\delta^2_{n+1} = \frac{\mu}{2} \frac{\sum_{(i,j)} (b^2_{n+1})_{i,j}[(D_x u_n)^2_{i,j} + (D_y u_n)^2_{i,j}]}{\sum_{(i,j)} (b^2_{n+1})_{i,j}}, \tag{3.2.3}$$

*where* $\mu \geq 1$ *is some parameter to be determined in advance based on noisy data* $f^{\epsilon}$*.*

*Step 3: Update image intensity* $u_{n+1}$*:*

$$u_{n+1} = u_n + \alpha_{n+1} v_{n+1}, \ n \geq 0, \tag{3.2.4}$$

*where the search direction $v_{n+1}$ is defined via*

$$\begin{aligned} \boldsymbol{v}_0 &= 0 \\ \boldsymbol{v}_{n+1} &= -\boldsymbol{g}_{n+1} + \frac{\boldsymbol{g}_{n+1}^T \boldsymbol{g}_{n+1}}{\boldsymbol{g}_n^T \boldsymbol{g}_n} \boldsymbol{v}_n, \ n \geq 0 \end{aligned} \tag{3.2.5}$$

*with the gradient $\boldsymbol{g}_{n+1}$ being*

$$\boldsymbol{g}_{n+1} = \boldsymbol{K}^T(\boldsymbol{K}\boldsymbol{u}_n - \boldsymbol{f}^\epsilon) + \frac{F(u_n)}{F_{n+1}^{multi}(u_n)} \boldsymbol{D}^T \boldsymbol{B}_{n+1} \boldsymbol{D}\boldsymbol{u}_n, \ n \geq 0 \tag{3.2.6}$$

*The step size $\alpha_{n+1}$ is chosen for each given search direction $v_{n+1}$ and image intensity $u_n$, as the real root of*

$$\frac{\partial}{\partial \alpha} C_{n+1}(u_n + \alpha v_{n+1})\big|_{\alpha=\alpha_{n+1}} = 0, \tag{3.2.7}$$

*thus ensuring that the cost functional $C_{n+1}(u_n + \alpha v_{n+1})$ in (3.2.1) reaches one of its stationary points as a polynomial in $\alpha$ (see Section 3.2.4 for details).*
*Termination criterion: If*

$$||u_{n+1} - u_n||_2^2 < tol \tag{3.2.8}$$

*then stop.*

To simplify calculations, let us use the following notation for the rest of this section: consider re-shaping the 2D image intensities $u_n$ and $f^\epsilon$ from $N \times N$ matrices to 1D arrays $\mathbf{u}_n$ and $\mathbf{f}^\epsilon$ with dimension $1 \times N^2$. In (3.2.6) $\mathbf{K}$, $\mathbf{D}$ and $\mathbf{B}_{n+1}$ are all $N^2 \times N^2$ matrices. They represent the convolution operator $\mathcal{K}$, finite difference operator $D_x$, $D_y$ and the weight function $b_{n+1}^2$ correspondingly in the re-shaped setting. Gradient $\mathbf{g}_n$, $\mathbf{g}_{n+1}$ and search directions $\mathbf{v}_n$, $\mathbf{v}_{n+1}$ are also $1 \times N^2$ arrays in (3.2.5). Specifically, matrix $\mathbf{B}_{n+1}$ takes the form of

$$\mathbf{B}_{n+1} = diag\{(b_{n+1}^2)_k, \ k = 1, 2, ...N^2\},$$

where entries $(b_{n+1}^2)_k$ corresponds to $(b_{n+1}^2)_{i,j},\ 1 \le i, j \le N$ in (3.2.2).

The above Algorithm 3.2.1 gives a big picture of the multiplicative approach. In the following subsections, I will fill in various details of this algorithm, which include (a) how to choose the initial value for $\delta_0^2$ in Step 0; (b) the motivation and derivation of the two recurrence relations (3.2.2) and (3.2.3) for $b_n^2$ and $\delta_n^2$ in Step 1 and 2; (c) how to find the step size $\alpha_{n+1}$ and the search direction $v_{n+1}$ in Step 3.

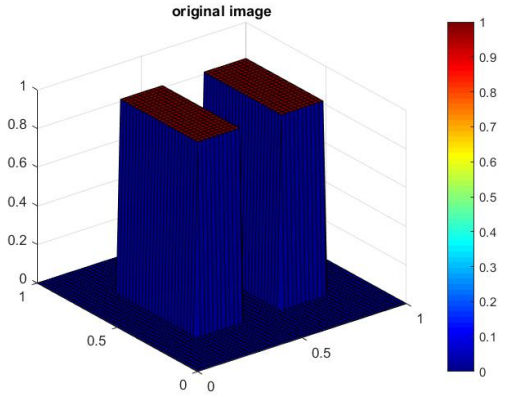
It is worth pointing out that the parameter $\mu$ does not exist in the recurrence relation in Step 2 in the original model proposed by Abubakar *et al.* The lacking of this parameter means that the model effectively takes $\mu = 1$ by default. In my research, I discovered that the numerical reconstruction was not satisfactory with a naive choice of $\mu = 1$, and could be improved significantly by inserting this parameter and increasing its value. Taking the exact image of (1.3.2) as an example, the reconstructed image with different choices of $\mu$ is illustrated in Fig 3.1 below:

It is observed that the insertion of parameter $\mu$ in (3.2.3) is essential. As the parameter $\mu$ plays a vital role in sharpening the reconstruction for piecewise constant images, it is crucial to make an appropriate choice for its value. In later Chapter 4, I will discuss in detail about how to choose its value appropriately, and present a straightforward choice rule for its value selection.

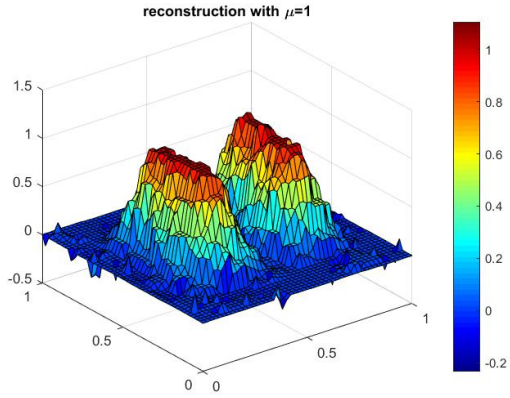
### 3.2.1 Recurrence Relation for $\delta_n^2$

The first recurrence relation (3.2.2) is motivated by the step size updating rule in (3.2.7). Consider in the simplified setting of noiseless case where $\eta = 0$, $f^\epsilon = f$ in (1.3.1), the image intensity $u(\mathbf{x})$ can be decomposed as a linear combination of the exact solution plus a perturbation in some generic direction:

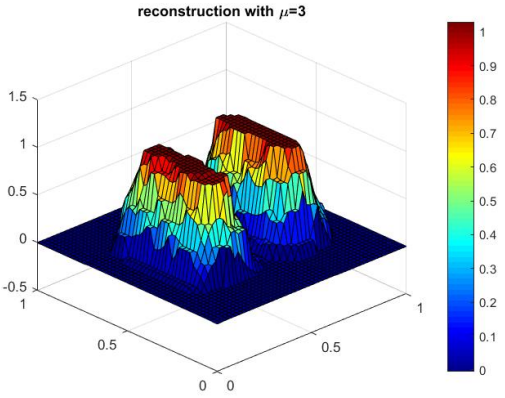
$$u = u^{exact} + \alpha v \tag{3.2.9}$$



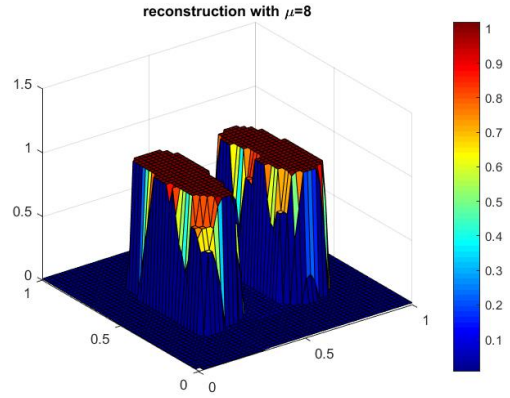

(a) exasct image




(b) reconstruction with Abubakar's original recurrence relation which corresponds to $\mu = 1$ in (3.2.3)




(c) reconstruction with $\mu = 3$ in (3.2.3)




(d) reconstruction with $\mu = 8$ in (3.2.3)

Figure 3.1: Original and reconstructed images with Abubakar's original recurrence relation and my modified recurrence relation (3.2.3) with different choices of $\mu$

Substitution of (3.2.9) into (3.2.1) gives rise to a polynomial of degree four in the step size $\alpha$:

$$C_{n+1}(u^{exact} + \alpha v) = \sum_{(i,j)} (\mathcal{K}v)_{i,j}^2 \alpha^2 (A\alpha^2 + 2B\alpha + C) \tag{3.2.10}$$

where

$$\begin{aligned} A &= \sum_{(i,j)} (b_{n+1}^2)_{i,j} [(D_x v)_{i,j}^2 + (D_y v)_{i,j}^2] \\ B &= \sum_{(i,j)} (b_{n+1}^2)_{i,j} [(D_x u^{exact})_{i,j} (D_x v)_{i,j} + (D_y u^{exact})_{i,j} (D_y v)_{i,j}] \\ C &= \sum_{(i,j)} (b_{n+1}^2)_{i,j} [(D_x u^{exact})_{i,j}^2 + (D_y u^{exact})_{i,j}^2 + \delta_{n+1}^2], \end{aligned} \tag{3.2.11}$$

with derivatives being

$$\begin{aligned} \frac{\partial C_{n+1}(u)}{\partial \alpha} &= 2 \sum_{(i,j)} (\mathcal{K}v)_{i,j}^2 \alpha (2A\alpha^2 + 3B\alpha + C) \\ \frac{\partial^2 C_{n+1}(u)}{\partial \alpha^2} &= 2 \sum_{(i,j)} (\mathcal{K}v)_{i,j}^2 (6A\alpha^2 + 6B\alpha + C) \end{aligned} \tag{3.2.12}$$

Abubakar *et al* [AvdBHB04] aim to find a condition for $\delta_{n+1}^2$ such that for each given search direction $v_{n+1}$, $C_{n+1}(u^{exact} + \alpha v)$ is convex in step size $\alpha$. This is equivalent to requiring its second order derivative (3.2.12) to be positive. Note that $\frac{\partial^2 C_n(u)}{\partial \alpha^2}$ is quadratic in $\alpha$ with leading coefficient $A \sum_{(i,j)} (\mathcal{K}v)_{i,j}^2 > 0$. Hence all we need is to have its discriminant negative, which means

$$\frac{B^2}{AC} < \frac{2}{3}. \tag{3.2.13}$$

By Schwartz Inequality applied to $A, B$ and $C$ as defined in (3.2.11), the following relation holds:

$$\frac{B^2}{A(C - \delta_{n+1}^2 \sum_{(i,j)} (b_{n+1}^2)_{i,j})} \leq 1. \tag{3.2.14}$$

Therefore, in order to fulfil condition (3.2.13), we need

$$\delta_{n+1}^2 \geq \frac{1}{2} \frac{\sum_{(i,j)} (b_{n+1}^2)_{i,j} [(D_x u^{exact})_{i,j}^2 + (D_y u^{exact})_{i,j}^2]}{\sum_{(i,j)} (b_{n+1}^2)_{i,j}}. \tag{3.2.15}$$

By replacing the unknown quantity $u^{exact}$ by $u_n$ and taking its lower bound, we get the recurrence relation (3.2.2) apart from the insertion of the tuning parameter $\mu$. The importance and methodology of choosing this parameter will be discussed in Chapter 4 in detail.

### 3.2.2 Recurrence Relation for $(b_n^2)_{i,j}$

The second recurrence relation (3.2.3) that defines $\delta_n^2$ and $(b_n^2)_{i,j}$ sequentially for all $1 \leq i, j \leq N$ is inspired by the following observation: As the number of iteration increases, the problem requires less and less regularisation strength, as the restored solution moves towards the exact solution more and more. This means that the updated $F_n^{multi}(u)$ should get closer to 1. In the spirit of this, the recurrence relation (3.2.3) in Step 1 is chosen to fit the regularisation term in such a way that

$$F_{n+1}^{multi}(u) = \frac{\sum_{(i,j)} (b_{n+1}^2)_{i,j} [(D_x u)_{i,j}^2 + (D_y u)_{i,j}^2 + \delta_{n+1}^2]}{\sum_{(i,j)} (b_{n+1}^2)_{i,j} [(D_x u_n)_{i,j}^2 + (D_y u_n)_{i,j}^2 + \delta_n^2]} \to 1, \quad n \to \infty$$

Note that this does not mean $u_n$ is directed to the un-regularised solution $[\mathcal{K}^*\mathcal{K}]^{-1}\mathcal{K}^* f^\epsilon$ as discussed in the previous section.

### 3.2.3 Choice of Initial Value $\delta_0^2$

In this section I will discuss the appropriate choice of initial value $\delta_0^2$ in the general framework of 3.2.1. This value induces all subsequent $\{b_n^2\}$ and $\{\delta_n^2\}$ for $n \geq 1$ iteratively, and may be chosen arbitrarily large with the following two reasons:

(i) It ensures the convexity of $C_1(u_0 + \alpha v_1)$ in $\alpha$ in the first iteration;

(ii) The multiplicative model is a self-adaptive regularisation method where the algorithm automatically determines "more important features" such as edges of the piecewise constant image, and takes special care of them by applying a different regularisation strength over them. This is controlled by the weight function $(b_n^2)_{i,j}$, which decides how much regularisation each pixel should have during each iteration, or in other words, where the "more important" pixels are. An arbitrarily large choice for $\delta_0^2$ outweighs the total variation of the corrupted image in the first iteration, *i.e.*

$$\frac{1}{N^2}[(D_x f^\epsilon)_{i,j}^2 + (D_y f^\epsilon)_{i,j}^2] << \delta_0^2,$$

and makes the weight function $(b_1^2)_{i,j}$ predominately constant for all pixels in this first iteration: $(b_1^2)_{i,j} \approx \frac{1}{\delta_0^2}$, $\forall\ i, j = 1, 2, ...N$, as can be seen from equation (3.2.2). This is a reasonable starting point, as there is no reason why we should favour certain pixels over the others at the start of the algorithm, if no prior information of the image is given. As we will see in later sections, the multiplicative regularisation appears to be rather stable and reliable for appropriate choices of parameter $\mu$, and appears to be much more sensitive to other parameters (e.g. the choice of $\mu$). Hence there is no urgent need to incorporate additional information of the image into $\delta_0^2$ considering the complexity of the model.

### 3.2.4 Search Direction and Step Size

In this subsection I will explain the determination of search direction $v_{n+1}$ and step size $\alpha_{n+1}$ in Step 3 of Algorithm 3.2.1. This step mimics a nonlinear conjugate gradient method of the Fletcher-Reeves type in (3.2.5) (see Section 2.3.3 for details), and the gradient direction $\mathbf{g}_{n+1}$ in (3.2.6) is calculated by considering the directional derivative of the objective cost functional. To determine the step size $\alpha_{n+1}$, consider the fourth degree polynomial $C_{n+1}(u_n + \alpha v_{n+1})$ in $\alpha$ (c.f. eq (3.2.1)), which, in the general case in the presence of noise $\eta$, takes the form

$$C_{n+1}(u_n + \alpha v_{n+1}) = (A\alpha^2 + 2B\alpha + C) \cdot (D\alpha^2 + 2E\alpha + F + G), \qquad (3.2.16)$$

with coefficients

$$\begin{aligned}
A &= \sum_{(i,j)} (\mathcal{K}v_{n+1})_{i,j}^2 \\
B &= \sum_{(i,j)} (\mathcal{K}v_{n+1})_{i,j} \sum_{(i,j)} [(\mathcal{K}u_n)_{i,j} - f_{i,j}^{\epsilon}] \\
C &= \sum_{(i,j)} [(\mathcal{K}u_n)_{i,j} - f_{i,j}^{\epsilon}]^2 \\
D &= \sum_{(i,j)} (b_{n+1}^2)_{i,j} [(D_x v_{n+1})_{i,j}^2 + (D_y v_{n+1})_{i,j}^2] \\
E &= \sum_{(i,j)} (b_{n+1}^2)_{i,j} [(D_x u_n)_{i,j} (D_x v_{n+1})_{i,j} + (D_y u_n)_{i,j} (D_y v_{n+1})_{i,j}] \\
F &= \sum_{(i,j)} (b_{n+1}^2)_{i,j} [(D_x u_n)_{i,j}^2 + (D_y u_n)_{i,j}^2] \\
G &= \delta_{n+1}^2 \sum_{(i,j)} (b_{n+1}^2)_{i,j},
\end{aligned}$$

and derivatives

$$\begin{aligned} & \frac{\partial}{\partial\alpha} C_{n+1}(u_n + \alpha v_{n+1}) \\ = \; & 2AD\alpha^3 + 3(AE + BD)\alpha^2 + (AF + AG + 4BE + CD)\alpha + (BF + BG + CE) \end{aligned} \tag{3.2.17}$$

and

$$\frac{\partial^2}{\partial\alpha^2} C_n(u_n + \alpha v_{n+1}) = 6AD\alpha^2 + 6(AE + BD)\alpha + (AF + AG + 4BE + CD).$$

Note that in the noiseless case as described in Section 4.2.1, the convexity of the cost functional $C_{n+1}(u^{exact} + \alpha v)$ in (3.2.10) is guaranteed by the recurrence relation (3.2.2). In this case $\alpha_n$ is chosen such that the global minimum of $C_{n+1}(u^{exact} + \alpha v)$ is reached in $\alpha$ for each given search direction $v_{n+1}$. Without explicit proof or clarification, Abubakar *et al* [AvdBHB04] simply extend the same result to the noisy case in Algorithm 3.2.1, and assume convexity of $C_{n+1}(u_n + \alpha v_{n+1})$ in (3.2.16). In the spirit of this, the cubic polynomial $\frac{\partial}{\partial\alpha} C_{n+1}(u_n + \alpha v_{n+1})$ in (3.2.17) in terms of $\alpha$ will be assumed to have one real root and a pair of roots being complex conjugate to each other, and step size $\alpha_{n+1}$ in Step 3 is updated as its only real root.

It is worth pointing out that despite a lack of solid theoretical backup, this updating rule for step size $\alpha_{n+1}$ appears to be fairly robust according to my numerical implementation as well as several implementations by Abubakar *et al* [AvdBHB04], [AvdB00], [AM02], [AvdB01]. In my opinion, a possible explanation lies in the semi-convergence nature of all multiplicative approaches. Formulated as a product of data fidelity term and some regularisation term, multiplicative regularisation is different from traditional regularisation methods which fulfils Definition 2.2.8. No explicit regularisation parameter is defined in the model. The fact that Abubakar *et al* does not enforce $u_{n+1}$ to be a minimiser of $C_{n+1}(u_n + \alpha v_{n+1})$ in Algorithm 3.2.1 Step 3 allows the solution to "stay away" from $[\mathcal{K}^*\mathcal{K}]^{-1}\mathcal{K}^* f^\epsilon$, which is effectively a direct inversion of problem (1.3.1) and cannot serve as an ideal approximation of the exact

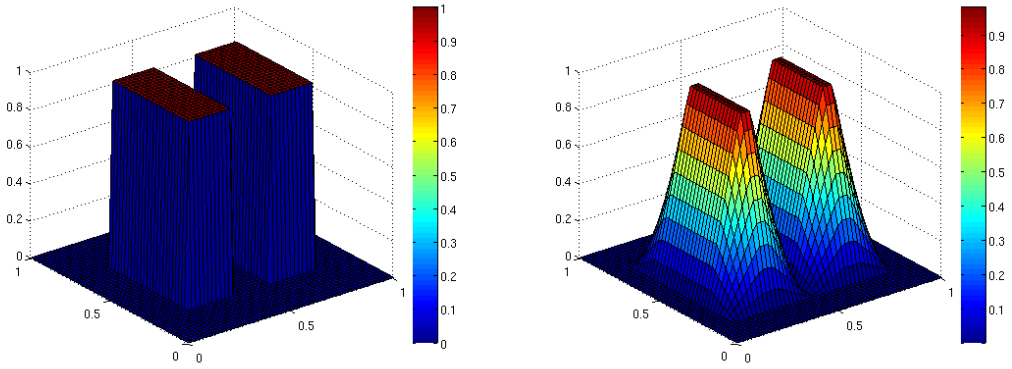
solution. In Chapter 5 of this dissertation, I will discuss multiplicative approaches in more general formulations, where $u_{n+1}$ will be updated via an inner loop regardless of the convexity of $C_{n+1}(u_n + \alpha v_{n+1})$, and I will present some theoretical results.

## 3.3 Numerical Results

In this section I will present numerical results obtained from the multiplicative regularisation algorithm (Algorithm 3.2.1). Given the two-block image (1.3.2) as the exact image $u^{exact}$, I will consider various symmetric and asymmetric distributions for noise $\eta$ in problem (1.3.1), and demonstrate advantages of the multiplicative approach from different perspective. I will use Gaussian noise as the natural representative for symmetrically distributed noise, since it is one of the most explored in the literature. K-distributed noise will be used as a representative of asymmetrically distributed noise, because it arises as a good model for the amplitude distribution in a wide variety of experiments involving scattering from turbulent media, such as the atmosphere and the sea (see, e.g. Jakeman and Pusey [JP76]); it also has a dependence on a "shape parameter" $\alpha$ which allows it to become close to two quite different types of asymmetric distributions, namely an exponential distribution for $\alpha < 1$, and a Rayleigh distribution for $\alpha > 1$ (see Appendix D for details and explicit formulae for the K-distribution). In either case, we use the standard deviation of the noise distribution as our metric for the noise level.

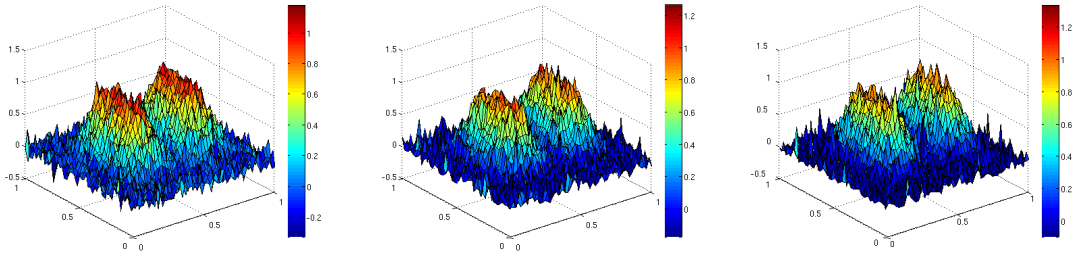
Let us start from the simplest piecewise constant test image (1.3.2) as the exact image $u^{exact}$ for problem (1.3.1). The exact and blurred image are presented in Fig (3.2) (a) and (b), with a choice of $\zeta = 0.1$ in the linear blurring kernel of (3.1.1). 10% noise of both symmetric (such as Gaussian) and asymmetric noise (different types of K-distribution as stated previously) is added to the blurred image in Fig 3.2(b). Corrupted test images are presented in Fig 3.3 (a)-(c) accordingly.

In order to interpret the multiplicative reconstructions better, I implement the



(a) exact test image (b) blurred image

Figure 3.2: exact and blurred test image



(a) 10% Gaussian noise (b) 10% K noise ($\alpha = 10$) (c) 10% K noise ($\alpha = 0.5$)

Figure 3.3: Corrupted test images with different noise distribution

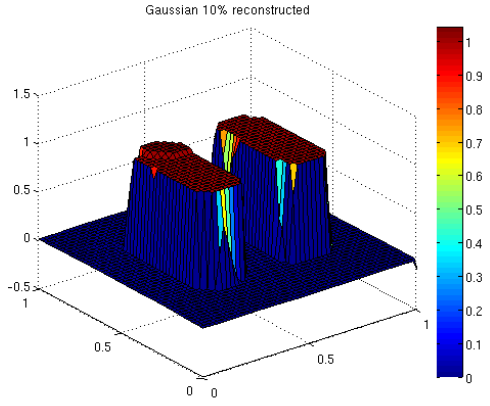
additive total variational regularisation as a reference. This requires the minimisation of $||\mathcal{K}u - f^{\epsilon}||_2^2 + \gamma^2 TV[u]$, where $TV[u]$ is the total variational regulariser as described in (2.1.2) , and $\gamma^2$ is determined by trial and error (in my case $\gamma^2 = 0.0006$ by trial and error). As we have discussed in previous Chapter 2.2.6, for additive type of models, the $L^2$ fidelity term $||\mathcal{K}u - f^{\epsilon}||_2^2$ is the most suitable for the Gaussian type of noise. This convex problem is solved by CVX package using Matlab as discussed in [GB14],[GB08].

| Noise type | Regularisation | SSIM | PSNR |
|---|---|---|---|
| Gaussian | Multiplicative | 0.68 | 18.76 |
| Gaussian | Additive | 0.75 | 21.58 |
| K ($\alpha = 10$) (asymm bell) | Multiplicative | 0.71 | 18.54 |
| K ($\alpha = 10$) (asymm bell) | Additive | 0.64 | 20.51 |
| K ($\alpha = 0.5$)(spiky) | Multiplicative | 0.70 | 19.64 |
| K ($\alpha = 0.5$)(spiky) | Additive | 0.64 | 21.24 |

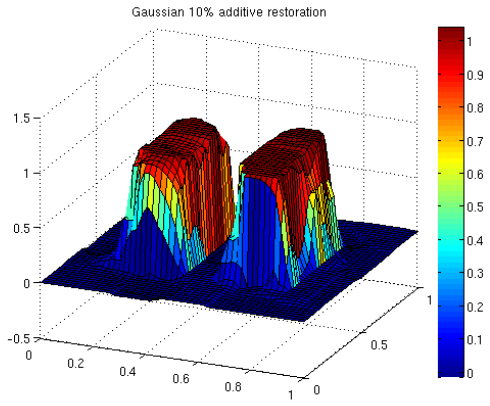
Table 3.1: Comparison of multiplicative and additive reconstructions with different noise distribution (All noise are at 10% level)

The quantitative and qualitative results for restoration of Fig 3.3 using both additive and multiplicative regularisation models are presented in Fig 3.4 and Table 3.1. I have used the PSNR and the SSIM as objective image quality metrics, since they are the most widely used. Neither can be always identified as best for assessing quality, since they are differently affected by the type of noise and deblurring, as well as the features of the images to be reconstructed. SSIM is better in general as a local measure, and as a measure of visual quality (see, e.g. Sheikh *et al* 2006 [SSB06], Chandler 2013 [Cha13]). It may be concluded that

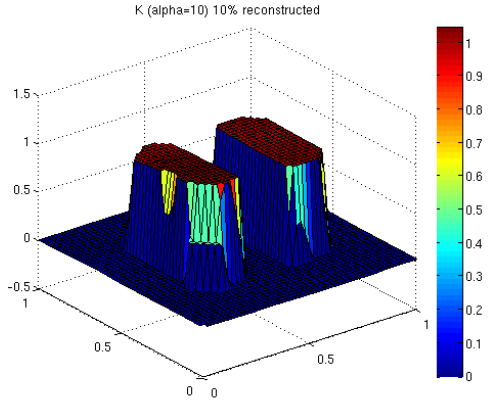
- The additive model behaves better in the Gaussian case than the K-distributions both qualitatively and quantitatively. This demonstrates the fact that $L^2$ fidelity term $||\mathcal{K}u - f^{\epsilon}||_2^2$ outperforms other choices of fidelity terms in the case of Gaussian noise in the additive regularisation setting (see Section 2.2.6). For the comparison across additive and multiplicative approach, it also may be



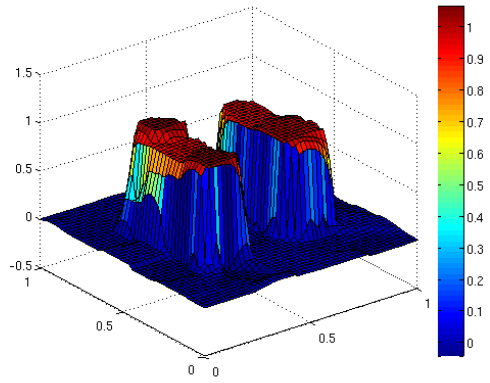

(a) multiplicative, 10% Gaussian



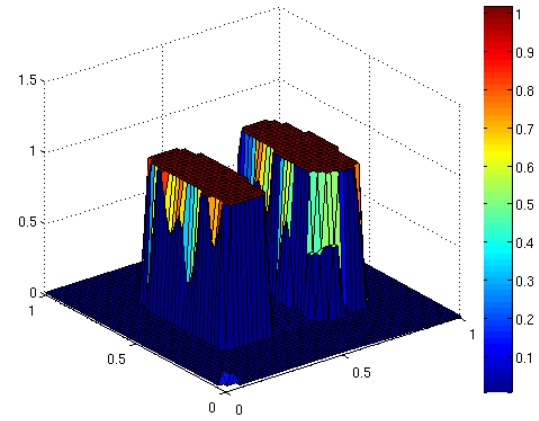

(b) additive, 10% Gaussian



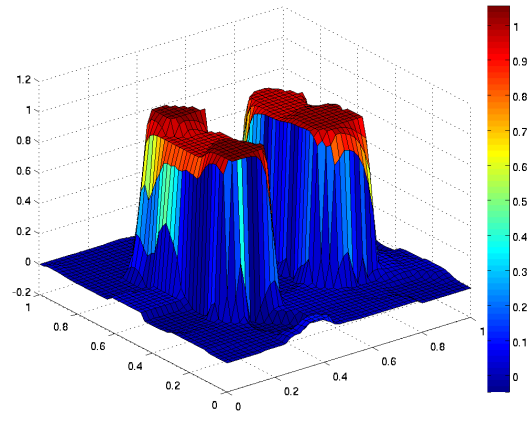

(c) multiplicative, 10% K noise ($\alpha = 10$)



(d) additive, 10% K noise ($\alpha = 10$)



(e) multiplicative, 10% K noise ($\alpha = 0.5$)



(f) additive, 10% K noise ($\alpha = 0.5$)

Figure 3.4: Reconstructed images from 10% noise by both multiplicative and additive regularisation models with different noise distribution

concluded that the multiplicative model does not suffer from the sensitivity of fidelity terms: Unlike its additive counterpart, there appears to be neither significant qualitative difference in the multiplicative reconstructions Fig 3.4 (a), (c) and (e), nor quantitative difference in the SSIM and PSNR indices in Table 3.1.

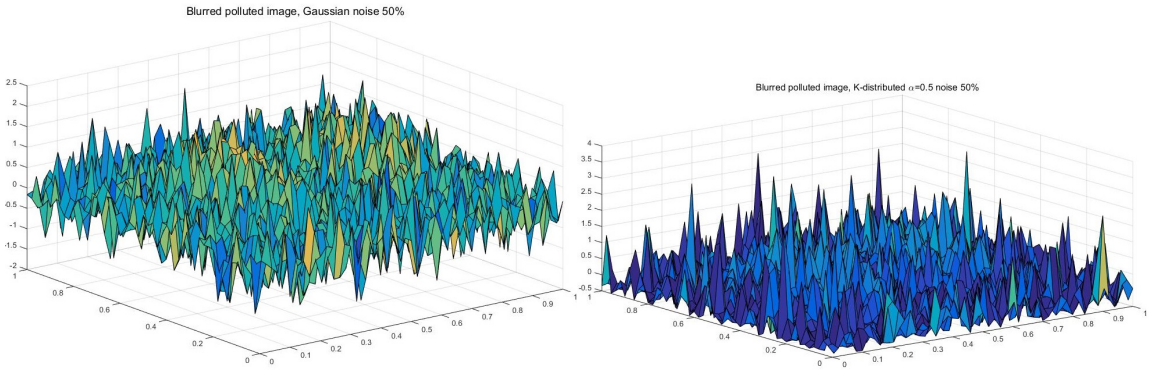
- Multiplicative regularisation performs similarly across all different types of noise tested, and outperforms additive regularisation for the two types of non-Gaussian noise. In this type of regularisation, there is no "clean" fidelity term, since $||\mathcal{K}u - f^{\epsilon}||_2^2$ is multiplied by $F_n^{multi}$, and is adapted at each step throughout the algorithm. Therefore the fidelity term is not related to the type of noise in the same way as it is in additive type of regularisation.

- The multiplicative model restores the main features of the image consistently better than its additive counterpart. This phenomenon is demonstrated in Fig 3.4, where the multiplicative reconstructions (a), (c) and (e) lead to parallel contours, and hence produce a "layer effect" for the piecewise constant blocks. This behavioural difference provides a strong evidence supporting the multiplicative model for piecewise constant images.

So far we have concluded that despite a compensation of smaller PSNR index which is based on residue errors, multiplicative reconstructions capture the overall image structure better than additive models. To further investigate this point, we impose large noise level to both regularisation models. At a noise level of 50%, the corruption effect is significant. Noisy images with different distributions are presented in Fig(3.5):

Reconstructions of 50% noise level with different regularisation models are shown in Fig 3.6, where it may be concluded that

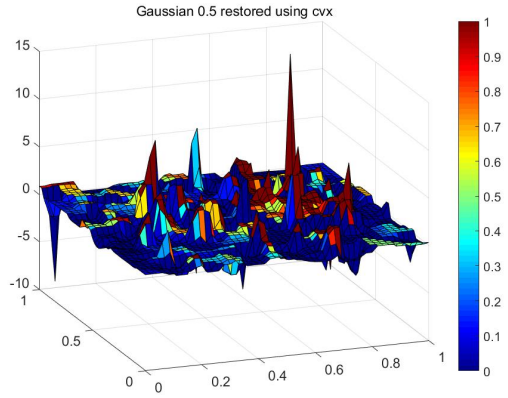
- The multiplicative model is able to cope with much higher noise levels than its additive counterpart, and still able to reconstruct the main features (*i.e.* the edges) when they are totally buried in the noisy data;



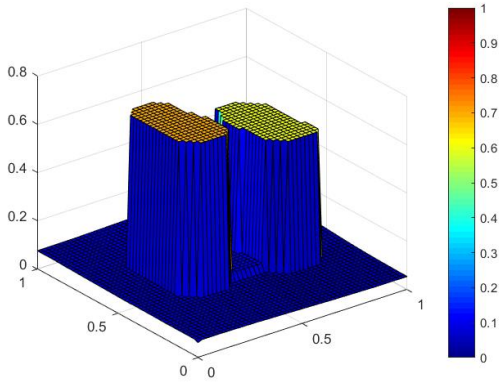

(a) Corrupted image with Gaussian 50% noise (b) Corrupted image with K ($\alpha = 0.5$) 50% noise (spiky)

Figure 3.5: 50% symmetric and asymmetric noise

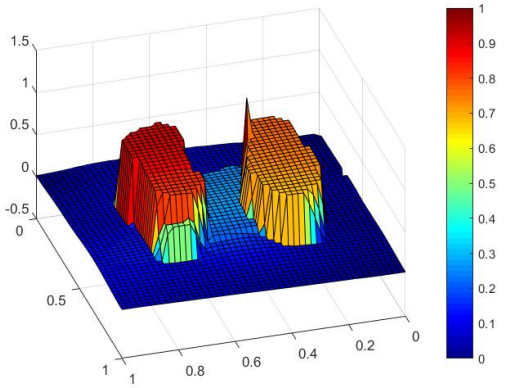
- Although the multiplicative model is robust with different noise distributions at high amplitude in terms of being able to reconstruct the image edges, its performance varies with different noise distributions, and in particular is less good in the case of a K distribution with small shape parameter $\alpha < 1$, where random spurious spikes may occur in the reconstruction. This is illustrated in Fig 3.6 (d) in the case of spiky noise with K distribution ($\alpha = 0.5$ ). However, this is expected and understandable: as K distributed noise with shape parameter $\alpha < 1$ is spiky on one end (see Fig D.1), in the case of high noise level up to 50%, this is a remarkably good performance.



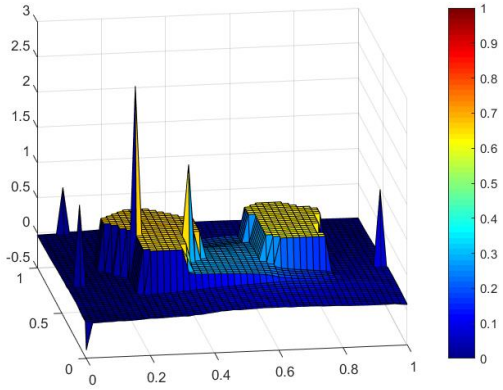

(a) Additive, Gaussian



(b) Multiplicative, Gaussian



(c) Multiplicative, K ($\alpha = 10$) (asymmetric bell)



(d) Multiplicative, K ($\alpha = 0.5$) (spiky)

Figure 3.6: Reconstructions at 50% noise level by different regularisation models for different noise distributions

# Chapter 4

# An Augmented Recurrence Relation

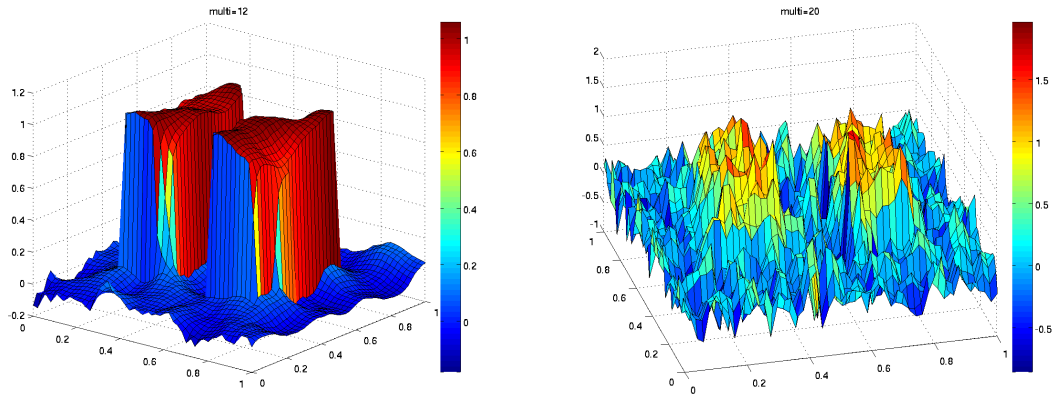
## 4.1 Motivation

In the previous chapter we have demonstrated the power of multiplicative regularisation in terms of the following aspects:

- Being robust over different noise distributions with the same $L^2$ data fidelity term applied;
- Being able to capture the main features of the image (e.g. the edges) and recover them in the case of large noise level;

It is worth pointing out that all the numerical results presented in Chapter 3 have the inserted tuning parameter $\mu$ appropriately chosen in the recurrence relation (3.2.3) This is my modification to the original multiplicative regularisation method by Abubakar *et al* [AvdBHB04]. The lack of this parameter is equivalent to taking $\mu = 1$ in (3.2.3). As Fig 3.1 has demonstrated, a very different choice of $\mu = 8$ is required for the model to realise its full potential. One might think that this satisfactory result is induced by a better geometry of the problem, in the sense that recurrence relation (3.2.3) now not only meets but also greatly exceeds the minimum

requirement for the objective functional $C_{n+1}(u_n + \alpha v_{n+1})$ in (3.2.16) to be convex in step size $\alpha$. Surprisingly this is not the whole story. The parameter $\mu$ has a much more profound meaning in adjusting the nature of the reconstruction. Funnily enough, there appears to be a narrow window where the choice of $\mu$ can make the algorithm work, just like the regularisation parameter $\alpha$ in additive types of regularisation. The following Fig (4.1) plus the previous results in Fig 3.1 demonstrate this point:




(a) $\mu = 12$ (b) $\mu = 20$

Figure 4.1: Reconstructed images with different choice of $\mu$ using the multiplicative regularisation model (from 10% Gaussian noise)

It shows from Fig 3.1 (a) that the result is under-regularised (pyramids shape rather than rectangles) for small choices of $\mu$, while artificial effects step in and essentially ruin the reconstruction totally with large values of $\mu$ (see Fig (4.1) (a) and (b). This suggests that $\mu$ plays a similar role as the regularisation parameter in traditional additive types of regularisation methods. Its value needs to be appropriately determined and only sensible choices may lead to satisfactory reconstructions.

This leads to a fundamental paradox of the model: the multiplicative method is motivated as a self-adapted regularisation to avoid the tuning of the regularisation parameter, which is in general computationally costly to determine. However, with

my modification of inserting an extra parameter $\mu$ in recurrence relation (3.2.3), this motivation seems to be pointless. To rectify this shortfall, I will show in the following chapter that with a little prior information of the image, it is possible to derive an explicit formula for an appropriate choice $\mu^*$, which will allow a straightforward determination of parameter $\mu$ significantly less expensive than the tuning of the regularisation parameter in its additive counterparts. With the addition of this choice rule, it is now possible for Abubakar's multiplicative regularisation model to obtain good results and at the same time avoid a parameter whose tuning requires large computational resources.

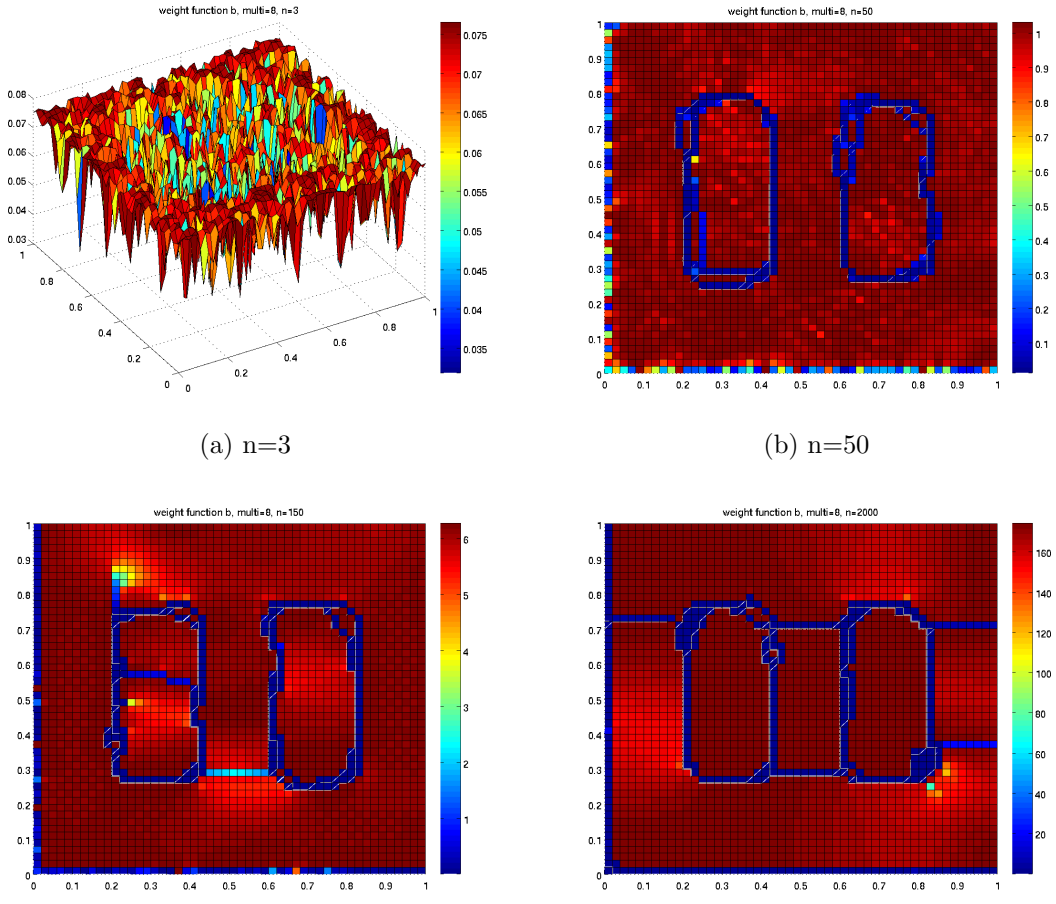
## 4.2 Insertion of Parameter $\mu$

### 4.2.1 A Necessary Condition

Before introducing the selection rule for tuning parameter $\mu$, in this section, I will present a necessary but not sufficient condition for the multiplicative approach to produce a satisfactory solution, which will be used in later sections to derive a straightforward parameter selection rule for $\mu$.

To figure out this necessary condition, firstly we consider the case in Fig 3.1 (d) where a good choice of $\mu = 8$ leads to satisfactory reconstruction. Fig 4.2 shows its corresponding evolution of weight function $b_n$, from which we can tell that $b_n$ eventually grows out of bound, despite the fact that its shape more or less converges to the shape of $(D_x u^{exact})^2_{i,j} + (D_y u^{exact})^2_{i,j}$ pixel-wisely.

This phenomenon is induced by recurrence relation (3.2.2) which enforces $(b_n^2)_{i,j} = \frac{1}{(D_x u_{n-1})^2_{i,j} + (D_y u_{n-1})^2_{i,j} + \delta^2_{n-1}} \; \forall \, i, j = 1, 2, ...N$.. For small values of $n$, $u_{n-1} \approx f^\epsilon$, and $\delta^2_{n-1} >> (D_x u_{n-1})^2_{i,j} + (D_y u_{n-1})^2_{i,j}$ due to initialization as discussed in Section 3.2.3. Hence $(b_n)_{i,j}$ is more evenly distributed over different pixels as illustrated in Fig 4.2 (a); As the number of iterations $n$ increases, the image reconstruction $u_n$ is sharpened further, which leads to a better distinction between "edge pix-




(a) n=3

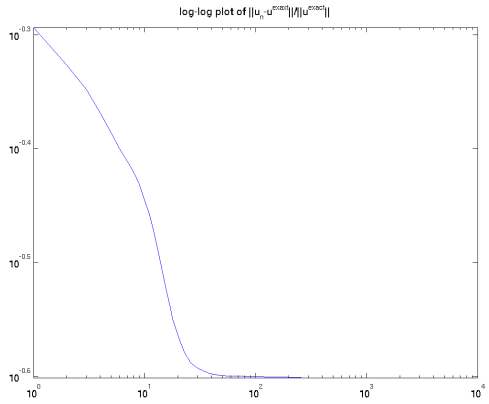
(b) n=50

(c) n=150

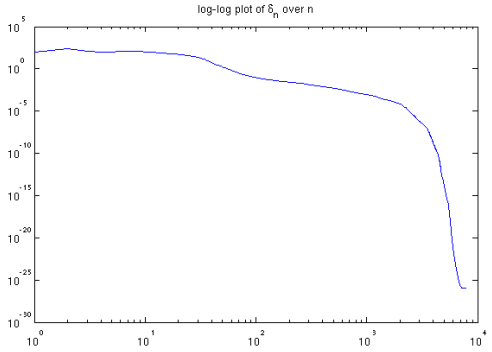
(d) n=2000

Figure 4.2: evolution of $(b_n)_{i,j}$, with 10% Gaussian noise and $\mu = 8$, $h = 0.02$

els" and "interior pixels" (i.e. not on the edges) for piecewise constant images. This eventually causes $\delta_{n-1}^2 << (D_x u_{n-1})_{i,j}^2 + (D_y u_{n-1})_{i,j}^2$ for "edge pixels" and $\delta_{n-1}^2 >> (D_x u_{n-1})_{i,j}^2 + (D_y u_{n-1})_{i,j}^2$ for "interior pixels" in recurrence relation (3.2.3), which leads to the convergence of weight function $b_n(\mathbf{x})$ in shape. The blowing up in scale of $b_n$ is due to the convergence of parameter $\delta_n \to 0$, as $n \to \infty$. This point is demonstrated in Fig 4.3 (b). In this case, both $(D_x u_{n-1})_{i,j}^2 + (D_y u_{n-1})_{i,j}^2 \to 0$ and $\delta_{n-1}^2 \to 0$ at "interior pixels" of the reconstructed image, which results in $(b_n)_{i,j} \to \infty$ at these pixels due to recurrence relation (3.2.2).



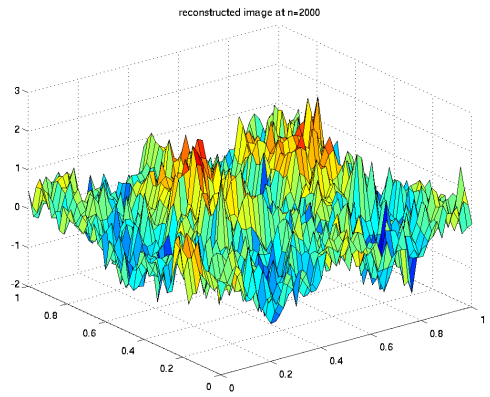
(a) log-log plot of $||u_n - u_{exact}||_2/||u_{exact}||_2$



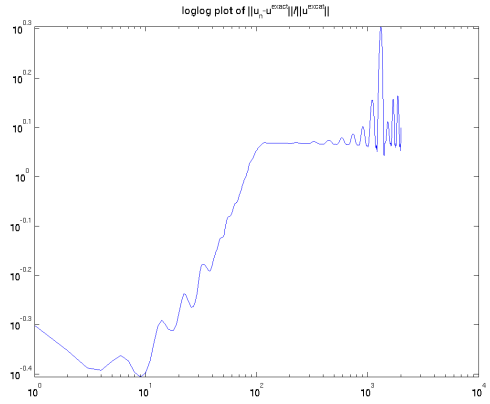

(b) log-log plot of $\delta_n^2$

Figure 4.3: evolution of residue error $||u_n - u_{exact}||_2/||u_{exact}||_2$ and $\delta_n^2$, with 10% Gaussian noise and $\mu = 8$, $h = 0.02$

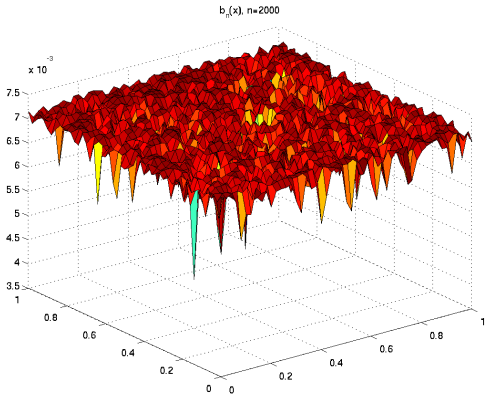
So far we have investigated the evolution of complementary variables $b_n$ and $\delta_n$ in the cases where the multiplicative regularisation leads to satisfactory reconstructions. It is then natural to wonder what would happen in the cases where the multiplicative regularisation fails to provide a satisfactory reconstruction. The results in Fig 4.4 shows how the method behaves with $\mu = 33$ and mesh size $h = 0.02$ for Gaussian 10% noise:



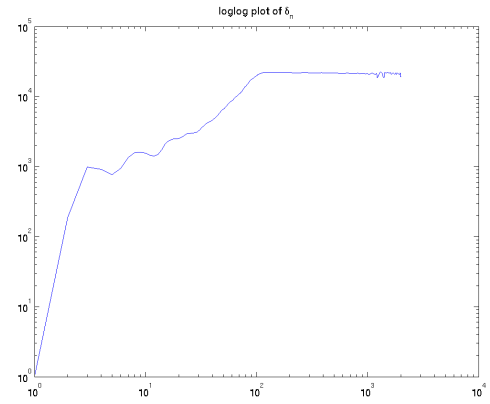

(a) reconstructed image with $\mu = 33$




(b) log-log plot of $||u_n - u_{exact}||_2/||u_{exact}||_2$




(c) weight function $b_n(\mathbf{x})$ at $n = 2000$




(d) log-log plot of $\delta_n^2$

Figure 4.4: results of $\mu = 33$ at $h = 0.02$ for Gaussian 10% noise

According to Fig 4.4 (a), this combination of parameter $\mu$ and mesh size $h$ results in ridiculous results, where large spurious oscillations occur throughout the reconstructed image. It also may be observed from Fig 4.4 (b) that the algorithm never converges nor to the correct solution: the parameter $\delta_n^2$ increases monotonically and stays at a large level as demonstrated in Fig 4.4 (d). As its direct consequence, the weight function $(b_n)_{i,j}$ stays small over all pixels, and hence fails to detect "edge pixels" over the "interior ones" in Fig 4.4 (c). This observation leads to a necessary but not sufficient condition, given a chosen mesh size, for the proper functioning of the multiplicative regularisation: $\delta_n^2$ should remain bounded as $n \to \infty$. To see this, we consider the physical meaning of recurrence relations (3.2.2) and (3.2.3): the right hand side of (3.2.3) is essentially an average weighted sum of finite differences $(D_x u_n)_{i,j}^2 + (D_y u_n)_{i,j}^2$ with weight $(b_{n+1})_{i,j}$ allocated to each pixel. As a decreasing function in terms of finite difference $(D_x u_n)_{i,j}^2 + (D_y u_n)_{i,j}^2$ by construction, the weight function $(b_{n+1})_{i,j}$ in 3.2.2 allocates less weight to "edge pixels" with large finite difference $(D_x u_n)_{i,j}^2 + (D_y u_n)_{i,j}^2$, but heavier weight to "interior pixels" with smaller finite difference $(D_x u_n)_{i,j}^2 + (D_y u_n)_{i,j}^2$. Hence it may be concluded that

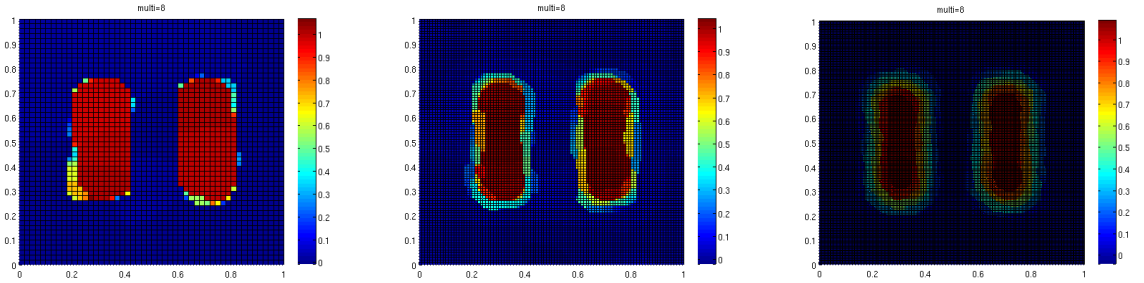
$$\delta_{n+1}^2 \leq \frac{\mu}{2N^2} \sum_{(i,j)} [(D_x u_n)_{i,j}^2 + (D_y u_n)_{i,j}^2], \tag{4.2.1}$$

where the right hand side is equivalent to setting the weight function $(b_{n+1})_{i,j}$ constant over all pixels $1 \leq i, j \leq N$. In the case where the multiplicative algorithm produces a satisfactory reconstruction, one would expect $u_n$ tend to the exact image $u^{exact}$. Replacing $u_n$ by $u^{exact}$ in (4.2.1) gives an upper bound for $\{\delta_n^2\}_{n \in \mathbb{N}}$. In later sections of this chapter, the formula for a straightforward choice of parameter $\mu$ is derived using the expressions derived for $\delta_n^2$ as $n$ increases.

### 4.2.2 Dependence on Mesh Size $h$

In this section I continue to explore possible factors that can influence the quality of reconstruction or the appropriate choice for $\mu$. An interesting feature of $\mu$ is, unlike

the case of additive regularisation parameter, its appropriate choice $\mu^*$ has some dependence on the mesh size $h^2$. To see this, we fix $\mu = 8$ and vary the mesh size $h^2$ in Fig (4.5):




(a) $h^2 = 0.02^2$, $||u_n - u^{exact}||_2/||u^{exact}||_2$=0.227

(b) $h^2 = 0.01^2$, $||u_n - u^{exact}||_2/||u^{exact}||_2$=0.315

(c) $h^2 = 0.005^2$, $||u_n - u^{exact}||_2/||u^{exact}||_2$=0.365

Figure 4.5: Reconstructed images with different mesh size $h^2$ at $\mu = 8$

It is observed that the choice $\mu = 8$ is good for mesh size $h^2 = 0.02^2$, but not for the other two mesh sizes as the residue error $||u_n - u^{exact}||_2/||u^{exact}||_2$ keeps increasing. In Fig (4.5) (b) and (c), the choice of $\mu$ appears to be too small, and the edges of the rectangular blocks are smoothed out. However, by increasing the value of $\mu$, reconstructions using these two mesh sizes may be improved as shown in Fig (4.6) and (4.7):

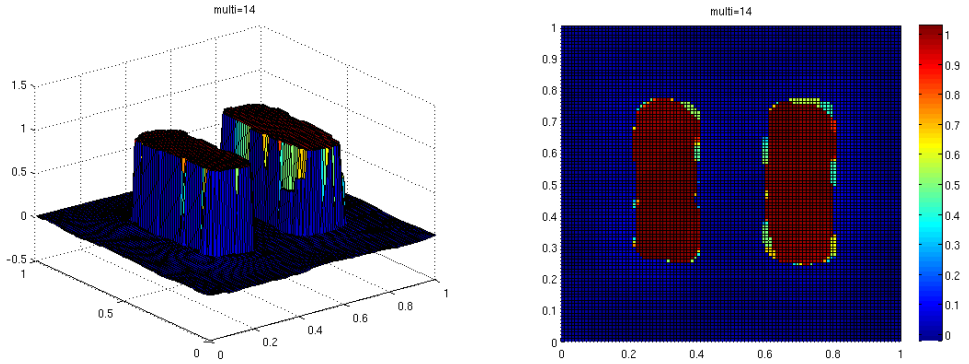



Figure 4.6: improved reconstructed images with $h^2 = 0.01^2$, $\mu = 14$, $||u_n - u^{exact}||/||u^{exact}||$=0.275

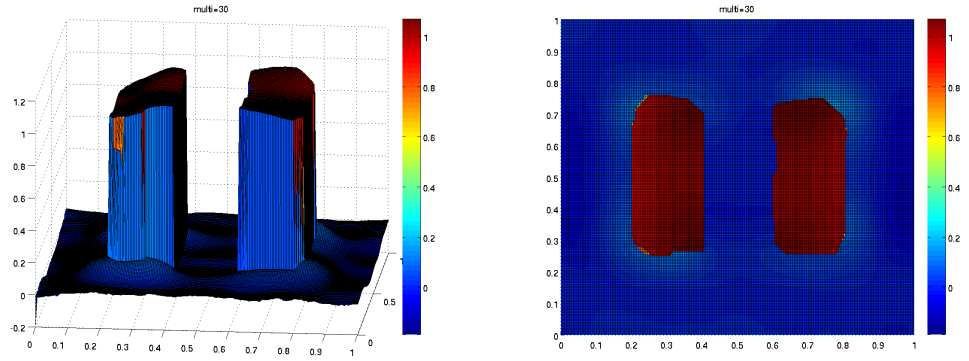



Figure 4.7: improved reconstructed images with $h^2 = 0.005^2$, $\mu = 20$, $||u_n - u^{exact}||/||u^{exact}||$=0.292

To understand why different mesh sizes $h$ would lead to different reconstructions and further affect the appropriate choice for $\mu$, let us consider recurrence relation (3.2.3) using coarse grids $N \times N$ (mesh size $h = \frac{1}{N}$) and fine grids $2N \times 2N$ (mesh size $\frac{h}{2} = \frac{1}{2N}$). Discretisation using fine grids (assuming $(b^2_{n+1})_{i,j}$ $(u_n)_{i,j}$ are $2N \times 2N$ matrices) gives rise to expression

$$\delta^2_{n+1}\big|_{N\times N} = \frac{\mu}{2}\frac{\sum_{1\le i,j\le N}(b^2_{n+1})_{2i,2j}[\big(\frac{u_{2i,2j}-u_{2i-2,2j}}{h}\big)^2+\big(\frac{u_{2i,2j}-u_{2i,2j-2}}{h}\big)^2]}{\sum_{1\le i,j\le N}(b^2_{n+1})_{2i,2j}}, \quad (4.2.2)$$

for coarse grids $N \times N$ (mesh size being $h = \frac{1}{N}$), and

$$\delta^2_{n+1}\big|_{2N\times 2N} = \frac{\mu}{2}\frac{\sum_{1\le i,j\le N}\sum_{2i-1\le k\le 2i,2j-1\le l\le 2j}(b^2_{n+1})_{k,l}[\big(\frac{u_{k,l}-u_{k-1,l}}{\frac{h}{2}}\big)^2+\big(\frac{u_{k,l}-u_{k,l-1}}{\frac{h}{2}}\big)^2]}{\sum_{1\le i,j\le N}\sum_{2i-1\le k\le 2i,2j-1\le l\le 2j}(b^2_{n+1})_{k,l}}, \quad (4.2.3)$$

for fine grids $2N \times 2N$ (mesh size being $\frac{h}{2} = \frac{1}{2N}$). The character $u$ in expression (4.2.2) and (4.2.3) represents the nth iterate $u_n$. Consider a $3 \times 3$ grid area where $2i-2 \le k \le 2i, 2j-2 \le l \le 2j$ for some $k, l \in \{1, 2, ...N\}$. If the image intensity $u$ is linearly interpolated over these 9 grids, then the following relationship regarding finite differences may be established between a single pixel $(2i, 2j)$ in the coarse grid setting and $2 \times 2$ pixels $\{(k,l)\big|2i-1 \le k \le 2i, 2j-1 \le l \le 2j\}$ in the fine grid setting:

$$\frac{u_{k,l}-u_{k-1,l}}{\frac{h}{2}} = \frac{u_{2i,2j}-u_{2i-2,2j}}{h}, \quad \frac{u_{k,l}-u_{k,l-1}}{\frac{h}{2}} = \frac{u_{2i,2j}-u_{2i,2j-2}}{h},$$
$$\forall\ 2i-1 \le k \le 2i, 2j-1 \le l \le 2j \quad (4.2.4)$$

This further leads to equality of weight function $b_{n+1}$ using recurrence relation (3.2.2):

$$(b_{n+1})_{2i,2j} = (b_{n+1})_{2i-1,2j} = (b_{n+1})_{2i,2j-1} = (b_{n+1})_{2i-1,2j-1}$$

We can hence conclude that there is equal contribution in (4.2.2) from a single pixel $(2i, 2j)$ in the coarse grid setting, as the contribution in (4.2.3) from $2 \times 2$ pixels $(k, l)$ $(2i-1 \le k \le 2i, 2j-1 \le l \le 2j)$ in the fine grid setting.

However, this is not the case if a discontinuity occurs in this $3 \times 3$ grid area locally. For example, one can verify that in the case where $u_{2i-2,l} - u_{2i-1,l} = 0, u_{2i,l} =$

$1, \forall 2j-2 \leq l \leq 2j$, the contribution into $\delta^2_{n+1}\big|_{N\times N}$ and $\delta^2_{n+1}\big|_{2N\times 2N}$ are different from single pixel $(2i, 2j)$ versus $2\times 2$ pixels $(k, l)$ $(2i-1 \leq k \leq 2i, 2j-1 \leq l \leq 2j)$. In fact, the finite differences in the fine grid setting are larger due to smaller mesh size. This leads to smaller weights $(b_{n+1})_{k,l}, 2i-1 \leq k \leq 2i, 2j-1 \leq l \leq 2j$ and eventually causes smaller contribution into $\delta^2_{n+1}\big|_{2N\times 2N}$ in the fine grid setting. This means that, as mesh size $h$ reduces, if one wants to keep the same regularisation process (e.g. same sequence $\{\delta^2_n\}_{n\in\mathbb{N}}$), a larger value for $\mu$ will be needed to compensate the loss from finite differences, which explains the need for a higher value of $\mu$ for finer grids. The suitable value of $\mu$ will be determined in a straightforward manner in later sections, taking into account its mesh size dependence.

## 4.3 Approximated Recurrence Relation

In order to obtain a closed-form formula for the parameter $\mu$, in this section I derive an approximation for the recurrence relation (3.2.3). This expression requires some simplification before it can be made useful for the derivation of $\mu$: we will firstly get rid of the weight function $b_{n+1}$ in the recurrence relation (3.2.3) by substituting (3.2.2) into (3.2.3), and then obtain an asymptotic expansion with respect to the finite difference term $(D_x u_n)^2_{i,j} + (D_y u_n)^2_{i,j}$ for both numerator and denominator of (3.2.3) as a function of the steering parameter $\delta^2_n$ and the finite difference term $(D_x u_n)^2_{i,j} + (D_y u_n)^2_{i,j}$ only.

We start by introducing domain partition according to the finite difference of each pixel. At $n$th iteration, we have

$$\Omega_n = \{(i,j)\big|(D_x u_n)^2_{i,j} + (D_y u_n)^2_{i,j} > \delta^2_n, 1 \leq i, j \leq N\}$$

and its complement $\Omega^c_n = \Omega \setminus \Omega_n$

$$\Omega^c_n = \{(i_c,j_c)\big|(D_x u_n)^2_{i_c,j_c} + (D_y u_n)^2_{i_c,j_c} \leq \delta^2_n, 1 \leq i_c, j_c \leq N\}$$

In other words, $\Omega_n$ contains all pixels whose finite differences $(D_x u_n)^2_{i,j} + (D_y u_n)^2_{i,j}$ is larger than $\delta_n^2$, while its complement $\Omega_n^c$ contains the rest of pixels whose finite differences are smaller than the threshold $\delta_n^2$. This definition differentiates "edge pixels" in $\Omega_n$ from "interior pixels" in $\Omega_n^c$ of the $n$th processed image with $\delta_n^2$ being the reference. We use $|\cdot|$ to denote the cardinality of set $\Omega$, $\Omega_n$ and $\Omega_n^c$, *i.e.* the number of pixels in these sets. Without loss of generality, we will assume

$$h^2|\Omega| = h^2(|\Omega_n^c| + |\Omega_n|) = 1, \forall\ n \in \mathbb{N} \tag{4.3.1}$$

in the rest of this thesis. The separation of pixels leads to different asymptotic expansions with respect to the finite difference term $(D_x u_n)^2_{i,j} + (D_y u_n)^2_{i,j}$ in different regions $\Omega_n$ and $\Omega_n^c$. For "edge pixels" in $\Omega_n$ where $(D_x u_n)^2_{i,j} + (D_y u_n)^2_{i,j} > \delta_n^2$:

$$\begin{aligned} & \frac{1}{(D_x u_n)^2_{i,j} + (D_y u_n)^2_{i,j} + \delta_n^2} \\ = \ & \frac{1}{(D_x u_n)^2_{i,j} + (D_y u_n)^2_{i,j}} \cdot \frac{1}{1 + \frac{\delta_n^2}{(D_x u_n)^2_{i,j} + (D_y u_n)^2_{i,j}}} \\ = \ & \frac{1}{(D_x u_n)^2_{i,j} + (D_y u_n)^2_{i,j}} - \frac{\delta_n^2}{[(D_x u_n)^2_{i,j} + (D_y u_n)^2_{i,j}]^2} + ..., \quad \forall (i,j) \in \Omega_n \end{aligned} \tag{4.3.2}$$

For "interior pixels" in $\Omega_n^c$ where $(D_x u_n)^2_{i_c,j_c} + (D_y u_n)^2_{i_c,j_c} \leq \delta_n^2$:

$$\begin{aligned} & \frac{1}{(D_x u_n)^2_{i_c,j_c} + (D_y u_n)^2_{i_c,j_c} + \delta_n^2} \\ = \ & \frac{1}{\delta_n^2} \cdot \frac{1}{1 + \frac{(D_x u_n)^2_{i_c,j_c} + (D_y u_n)^2_{i_c,j_c}}{\delta_n^2}} \\ = \ & \frac{1}{\delta_n^2} - \frac{(D_x u_n)^2_{i_c,j_c} + (D_y u_n)^2_{i_c,j_c}}{\delta_n^4} + ..., \quad \forall (i_c,j_c) \in \Omega_{n-1}^c \end{aligned} \tag{4.3.3}$$

We need to check the relative magnitude of the terms in (4.3.2) and (4.3.3), to ensure that we keep all relevant terms. First, recall that $\frac{\delta_n^2}{(D_x u_n)^2_{i,j} + (D_y u_n)^2_{i,j}} < 1$ in $\Omega_n$, and $\frac{(D_x u_n)^2_{i_c,j_c} + (D_y u_n)^2_{i_c,j_c}}{\delta_n^2} < 1$ in $\Omega_n^c$. To compare the magnitude of terms across expression

(4.3.2) and (4.3.3) over different regions, we further assume these two terms are of same order, *i.e.*,

$$1 >> \frac{\delta_n^2}{(D_x u_n)_{i,j}^2 + (D_y u_n)_{i,j}^2} \approx \frac{(D_x u_n)_{i_c,j_c}^2 + (D_y u_n)_{i_c,j_c}^2}{\delta_n^2},\ \forall (i,j) \in \Omega_n,\ (i_c, j_c) \in \Omega_n^c, \quad (4.3.4)$$

These are reasonable assumptions, given the definition of $\delta_n$, and are supported by numerical results. They give rise to the following ordering of the first two terms in asymptotic expansions (4.3.2) and (4.3.3):

$$\frac{1}{\delta_n^2} >> \frac{(D_x u_n)_{i_c,j_c}^2 + (D_y u_n)_{i_c,j_c}^2}{\delta_{n-1}^4} \approx \frac{1}{(D_x u_n)_{i,j}^2 + (D_y u_n)_{i,j}^2} >> \frac{\delta_n^2}{[(D_x u_n)_{i,j}^2 + (D_y u_n)_{i,j}^2]^4},$$
$$\forall (i,j) \in \Omega_n,\ (i_c, j_c) \in \Omega_n^c \quad (4.3.5)$$

This relationship will be used in a later derivation to compare the magnitude of terms across different regions $\Omega_n$ and $\Omega_n^c$.

We can now derive an asymptotic expansion for $\sum_{(i,j)} (b_{n+1}^2)_{i,j}$. By inserting (4.3.2) and (4.3.3) into (3.2.2), we obtain

$$\begin{aligned}\sum_{(i,j)} (b_{n+1}^2)_{i,j} &= \sum_{(i,j)\in\Omega_n} \frac{1}{(D_x u_n)_{i,j}^2 + (D_y u_n)_{i,j}^2 + \delta_n^2} + \sum_{(i_c,j_c)\in\Omega_n^c} \frac{1}{(D_x u_n)_{i_c,j_c}^2 + (D_y u_n)_{i_c,j_c}^2 + \delta_n^2} \\ &\approx \frac{1}{\delta_n^2}(1 - h^2|\Omega_n|) + \sum_{(i,j)\in\Omega_n} \frac{1}{(D_x u_n)_{i,j}^2 + (D_y u_n)_{i,j}^2} - \frac{1}{\delta_n^4} \sum_{(i_c,j_c)\in\Omega_n^c} [(D_x u_n)_{i_c,j_c}^2 + (D_y u_n)_{i_c,j_c}^2].\end{aligned} \quad (4.3.6)$$

The approximation in the last step of (4.3.6) comes from omitting terms at similar order as $\frac{\delta_n^2}{[(D_x u_n)_{i,j}^2 + (D_y u_n)_{i,j}^2]^4}$ in (4.3.5) and above. For the remaining terms in (4.3.6),

we have the following ordering:

$$\frac{1}{\delta_n^2}(1-h^2|\Omega_n|) >> \sum_{(i,j)\in\Omega_n}\frac{1}{(D_xu_n)_{i,j}^2+(D_yu_n)_{i,j}^2}-\frac{1}{\delta_n^4}\sum_{(i_c,j_c)\in\Omega_n^c}[(D_xu_n)_{i_c,j_c}^2+(D_yu_n)_{i_c,j_c}^2]. \tag{4.3.7}$$

In other words, in expression (4.3.6), the first term is the leading term, and both the second and the third terms are first order terms in the asymptotic expansion.

After obtaining the asymptotic expansion for the denominator of recurrence relation (3.2.2), we can apply the same strategy to its numerator $\sum_{(i,j)}(b_{n+1}^2)_{i,j}[(D_xu_n)_{i,j}^2+(D_yu_n)_{i,j}^2]$. Let us start with approximating $\frac{|\nabla u_{n-1}(\mathbf{x})|^2|}{|\nabla u_{n-1}(\mathbf{x})|^2+\delta_{n-1}^2}$ in the two different regions $\Omega_n$ and $\Omega_n^c$ separately:

For "edge pixels" in $\Omega_n$ where $(D_xu_n)_{i,j}^2+(D_yu_n)_{i,j}^2>\delta_n^2$:

$$\frac{(D_xu_n)_{i,j}^2+(D_yu_n)_{i,j}^2}{(D_xu_n)_{i,j}^2+(D_yu_n)_{i,j}^2+\delta_n^2}=1-\frac{\delta_n^2}{(D_xu_n)_{i,j}^2+(D_yu_n)_{i,j}^2}+..., \quad \forall(i,j)\in\Omega_n. \tag{4.3.8}$$

For "interior pixels" in $\Omega_n^c$ where $(D_xu_n)_{i_c,j_c}^2+(D_yu_n)_{i_c,j_c}^2\leq\delta_n^2$:

$$\frac{(D_xu_n)_{i_c,j_c}^2+(D_yu_n)_{i_c,j_c}^2}{(D_xu_n)_{i_c,j_c}^2+(D_yu_n)_{i_c,j_c}^2+\delta_n^2}=\frac{(D_xu_n)_{i_c,j_c}^2+(D_yu_n)_{i_c,j_c}^2}{\delta_n^2}-\frac{[(D_xu_n)_{i_c,j_c}^2+(D_yu_n)_{i_c,j_c}^2]^2}{\delta_n^4}+..., \quad \forall \quad (i_c,j_c)\in\Omega_n^c \tag{4.3.9}$$

Using assumption (4.3.4), it is possible to compare terms across different regions in expression (4.3.8) and (4.3.9) as follows:

$$1>>\frac{\delta_n^2}{(D_xu_n)_{i,j}^2+(D_yu_n)_{i,j}^2} \approx \frac{(D_xu_n)_{i_c,j_c}^2+(D_yu_n)_{i_c,j_c}^2}{\delta_n^2}>>\frac{[(D_xu_n)_{i_c,j_c}^2+(D_yu_n)_{i_c,j_c}^2]^2}{\delta_n^4}, \quad \forall \quad (i,j)\in\Omega_n,\ (i_c,j_c)\in\Omega_n^c, \tag{4.3.10}$$

Inserting (4.3.8) and (4.3.9)) in recurrence relation (3.2.3), we obtain an approxima-

tion for its numerator $\sum_{(i,j)}(b^2_{n+1})_{i,j}[(D_xu_n)^2_{i,j} + (D_yu_n)^2_{i,j}]$, which does not depend on $b^2_{n+1}$:

$$\begin{aligned}
&\sum_{(i,j)}(b^2_{n+1})_{i,j}[(D_xu_n)^2_{i,j} + (D_yu_n)^2_{i,j}] \\
= &\sum_{(i,j)\in\Omega_n} \frac{(D_xu_n)^2_{i,j} + (D_yu_n)^2_{i,j}}{(D_xu_n)^2_{i,j} + (D_yu_n)^2_{i,j} + \delta^2_n} + \sum_{(i_c,j_c)\in\Omega^c_n} \frac{(D_xu_n)^2_{i_c,j_c} + (D_yu_n)^2_{i_c,j_c}}{(D_xu_n)^2_{i_c,j_c} + (D_yu_n)^2_{i_c,j_c} + \delta^2_n} \\
\approx &\ 1 \cdot h^2|\Omega_n| - \delta^2_n \sum_{(i,j)\in\Omega_n} \frac{1}{(D_xu_n)^2_{i,j} + (D_yu_n)^2_{i,j}} + \frac{1}{\delta^2_n} \sum_{(i_c,j_c)\in\Omega^c_n} [(D_xu_n)^2_{i_c,j_c} + (D_yu_n)^2_{i_c,j_c}] \quad (4.3.11)
\end{aligned}$$

The approximation in the last step of (4.3.11) comes from omitting terms at similar order as $\frac{[(D_xu_n)^2_{i_c,j_c}+(D_yu_n)^2_{i_c,j_c}]^2}{\delta^4_n}$ in (4.3.10) and above. According to (4.3.10), we have the following ordering in magnitude for the terms in (4.3.11):

$$h^2|\Omega_n| >> -\sum_{(i,j)\in\Omega_n} \frac{\delta^2_n}{(D_xu_n)^2_{i,j} + (D_yu_n)^2_{i,j}} + \frac{1}{\delta^2_n} \sum_{(i_c,j_c)\in\Omega^c_n} [(D_xu_n)^2_{i_c,j_c} + (D_yu_n)^2_{i_c,j_c}] \quad (4.3.12)$$

In other words, in expression (4.3.11), the first term is the leading term, and both the second and the third terms are first order terms in the asymptotic expansion.

We can now derive an approximation for the recurrence relation (3.2.3) by substituting both expression (4.3.6) and (4.3.11) into the denominator and the numerator of (3.2.3):

$$\delta^2_{n+1} \approx \frac{\mu}{2} \frac{c_n - \delta^2_n b_n + \frac{a_n}{\delta^2_n}}{\frac{1}{\delta^2_n}(1 - c_n) + b_n + \frac{a_n}{\delta^4_n}} \quad (4.3.13)$$

where

$$\begin{aligned} a_n &= \sum_{(i,j)\in\Omega_n^c} [(D_x u_n)^2_{i_c,j_c} + (D_y u_n)^2_{i_c,j_c}] \\ b_n &= \sum_{(i,j)\in\Omega_n} \frac{1}{(D_x u_n)^2_{i,j} + (D_y u_n)^2_{i,j}} \\ c_n &= h^2|\Omega_n| \end{aligned}$$

## 4.4 Choice Rule for Parameter $\mu$

Given (4.3.13), in this section I can now derive an explicit formula for choosing $\mu$. Let us start from the following definitions based on the exact image intensity $u^{exact}$:

$$\begin{aligned} a &= \sum_{(i,j)\in\Omega_\infty^c} [(D_x u^{exact})^2_{i_c,j_c} + (D_y u^{exact})^2_{i_c,j_c}] \\ b &= \sum_{(i,j)\in\Omega_\infty} \frac{1}{(D_x u^{exact})^2_{i,j} + (D_y u^{exact})^2_{i,j}} \\ c &= h^2|\Omega_\infty|, \end{aligned} \tag{4.4.1}$$

where

$$\Omega_\infty = \{(i,j)\big|(D_x u^{exact})^2_{i,j} + (D_y u^{exact})^2_{i,j} > \delta_M^2, 1 \le i,j \le N\}$$

is the partition set of "interior" pixels of the discrete exact image. Here a rough estimation of $\delta_M^2$, where $M > 0$ is the termination index of the multiplicative regularisation (Algorithm 3.2.1), serves the purpose of providing a sensible separation between "interior" and "edge" pixels of the exact image intensity. Its actual value is not in itself of interest to us. In practice this separation may be obtained, for example, by applying a Sobel filter using an inbuilt automatic choice of threshold as discussed in [R20a] in the pre-processing step. Details will be further presented in Section 4.5.

We will assume the following relations regarding the quantities $a, b$ and $c$ by considering their physical meanings. As the pixel-wise sum of finite differences over set $\Omega_\infty^c$ of "interior pixels" of the exact image, quantity $a$ will be 0 in the case of pixel-wise constant images. Hence we assume its scale to be relatively small compared to the other two quantities $b$ and $c$, *i.e.*

$$0 < a << b,\ c \tag{4.4.2}$$

It follows from equation (4.3.1) that

$$0 < c < 1, \tag{4.4.3}$$

By approximating the quantities $a_n, b_n$ and $c_n$ by $a, b$ and $c$ defined above in (4.3.13), we now obtain

$$\delta_{n+1}^2 \approx \frac{\mu}{2} \frac{c - \delta_n^2 b + \frac{a}{\delta_n^2}}{\frac{1}{\delta_n^2}(1-c) + b + \frac{a}{\delta_n^4}}, \tag{4.4.4}$$

which depends only on the exact image $u^{exact}$, mesh size $h$, and implicitly some choice of $\delta_\infty^2$ as the threshold to distinguish "interior" and "edge" pixels of the exact image intensity. The ordering (4.3.7) and (4.3.12) in previous section leads to the following term ordering in the numerator and the denominator separately:

$$\begin{aligned} c &>> -\delta_n^2 b + \frac{a}{\delta_n^2}, \\ \frac{1}{\delta_n^2}(1-c) &>> b + \frac{a}{\delta_n^4}, \end{aligned} \tag{4.4.5}$$

We will make use of the approximated recurrence relation (4.4.4) for choosing an appropriate value for parameter $\mu$. A possible condition leading to a choice of which ensures good reconstruction, is to require that the sequence $\{\delta_n^2\}_{n\in\mathbb{N}}$ converges as $n \to \infty$. This is stronger than what we stated is required, but leads to a useful formula for choosing a $\mu$ that will give satisfactory reconstruction.

A naive approach from here is to take only the leading terms from both numerator and denominator of (4.4.4) using (4.4.5). This leads to the simple recurrence relation

$$\frac{\delta_n^2}{\delta_{n-1}^2} \approx \frac{\mu}{2} \cdot \frac{c}{1-c} \tag{4.4.6}$$

The condition that sequence $\{\delta_n^2\}_{n\in\mathbb{N}}$ converges is simply

$$\left| \frac{\mu}{2} \cdot \frac{c}{1-c} \right| < 1,$$

which is equivalently

$$\mu < 2\left(\frac{1}{c} - 1\right) \tag{4.4.7}$$

Unfortunately, equation (4.4.7) gives a bound that is far too wide and does not work well in practice. However, this bound is still useful, since it provides a third relation regarding quantities $a$, $b$, $c$ and $\mu$ in addition to (4.4.2) and (4.4.3), and we need to retain all terms in (4.4.4) in order to obtain a better formula for choosing $\mu$. By re-arranging (4.4.7), we obtain

$$2 - (\mu + 2)c > 0, \tag{4.4.8}$$

which will be used in later derivation.

Now, since we require convergence of $\{\delta_n^2\}_{n\in\mathbb{N}}$, we can assume $\delta_{n+1}^2 \approx \delta_n^2$ and set $x = \delta_{n-1}^2 = \delta_n^2$, obtaining the following auxiliary equation:

$$x = f(x) := \frac{\mu}{2} \frac{-bx^3 + cx^2 + ax}{bx^2 + (1-c)x + a}. \tag{4.4.9}$$

Solving this fixed point equation is equivalent to solving the quadratic equation

$$(1 + \frac{\mu}{2})bx^2 + (1 - c - \frac{\mu}{2}c)x + (1 - \frac{\mu}{2})a = 0, \tag{4.4.10}$$

which leads to roots as well as fixed point solution $f(x_i) = x_i$, $i = 1, 2$

$$x_{1,2} = \frac{-2 + (\mu+2)c \pm \sqrt{[2-(\mu+2)c]^2 - 4(\mu+2)(2-\mu)ab}}{2(\mu+2)b}$$

As this is too complicated an expression, we approximate both roots $x_1$ and $x_2$ as follows:

$$\begin{aligned} x_1 &= \frac{-2 + (\mu+2)c + \sqrt{[2-(\mu+2)c]^2 - 4(\mu+2)(2-\mu)ab}}{2(\mu+2)b} \\ &= \frac{2(\mu-2)a}{2 - (\mu+2)c + \sqrt{[2-(\mu+2)c]^2 - 4(\mu+2)(2-\mu)ab}} \\ &\approx \frac{(\mu-2)a}{2-(\mu+2)c} \end{aligned} \tag{4.4.11}$$

$$\begin{aligned} x_2 &= \frac{-2 + (\mu+2)c - \sqrt{[2-(\mu+2)c]^2 - 4(\mu+2)(2-\mu)ab}}{2(\mu+2)b} \\ &\approx \frac{-2 + (\mu+2)c - [2-(\mu+2)c]}{2(\mu+2)b} \\ &= -\frac{2-(\mu+2)c}{(\mu+2)b} \end{aligned} \tag{4.4.12}$$

The two approximation steps in derivation of (4.4.11) and (4.4.12) come from the following relation

$$[2-(\mu+2)c]^2 >> 4(\mu+2)(\mu-2)ab, \tag{4.4.13}$$

which may be justified by (1) applying relation (4.4.2) and (4.4.3) regarding quantities $a, b, c$ and $\mu$; and (2) additionally assuming quantity $2-(\mu+2)c$ is roughly $\mathcal{O}(1)$, *i.e.*

$$a << 2-(\mu+2)c \backsimeq \mathcal{O}(1). \tag{4.4.14}$$

This is easily verified in the case of piecewise constant image Fig 3.1, where $\mu = 8$

leads to a good reconstruction with mesh size $h = 0.02$. We have calculated that $c = 0.056$ in this case. In Section 4.5, I will use more complicated numerical examples to demonstrate that the $c$ value is small with general digital images, which further justifies assumption (4.4.14).

Let us now get back to the two roots $x_1$ and $x_2$ of the auxiliary equation (4.4.9) and analyse their property. As a direct consequence of (4.4.13), we have

$$[2 - (\mu + 2)c]^2 >> (\mu + 2)(\mu - 2)ab.$$

This is equivalent to

$$\frac{(\mu - 2)a}{2 - (\mu + 2)c} << \frac{2 - (\mu + 2)c}{(\mu + 2)b},$$

which leads to

$$|x_1| << |x_2|.$$

for the magnitude of roots $x_1$ and $x_2$. We choose the smaller fixed point $x_1$ as the solution used to derive a formula for choosing $\mu$. This choice relies on an argument based on perturbation theory. The fixed point equation (4.4.9), of second order in $x$, was derived from the recurrence relation (4.4.4) by requiring convergence of $\{\delta_n^2\}_{n\in\mathbb{N}}$. If we use the simplified version of (4.4.6) where only leading order terms are considered in both numerator and the denominator of (4.4.4), we obtain, under the same requirements used to derive (4.4.9), a fixed point equation of first order in $x$, with solution $x_0 = 0$. From a perturbation theory point of view, the smaller fixed point $x_1 \backsimeq 0$ is perturbed from $x_0 = 0$ directly, while $x_2$ is induced when $\delta_n^2$ gets larger.

In the spirit of this, we express $\delta_n^2 = x_i + \epsilon_n$, $i = 1, 2$ for sufficiently large $n$ at both fixed points $x_1$ and $x_2$. Linearisation gives rise to

$$x_i + \epsilon_n = f(x_i + \epsilon_{n-1}) = f(x_i) + f'(x_i)\epsilon_{n-1},$$

*i.e.*, since $x_i$ is a fixed point,

$$\epsilon_n = f'(x_i)\epsilon_{n-1}.$$

According to perturbation theory,

$$\begin{aligned} |f'(x_i)| < 1 &\iff x_i \textit{ is stable} \\ |f'(x_i)| > 1 &\iff x_i \textit{ is unstable.} \end{aligned}$$

From (4.4.9), we get:

$$f'(x) = \frac{\mu}{2}\frac{-b^2x^4 - 2b(1-c)x^3 + [c(1-c) - 4ab]x^2 + 2acx + a^2}{[a + (1-c)x + bx^2]^2}.$$

Using (4.4.11), we obtain the following approximation of $f'(x_1)$:

$$f'(x_1) \approx \frac{c\mu^2 - 4c\mu + 4(1-c)}{2\mu(1-2c)} \tag{4.4.15}$$

We choose $\mu$ such that not only $|f'(x_1)| < 1$, but also such that $|f'(x_1)|$ is minimised. We have:

$$\begin{aligned} |f'(x_1)| = \left|\frac{c\mu^2 - 4c\mu + 4(1-c)}{2\mu(1-2c)}\right| &= \left|\frac{1}{1-2c}\right|\left|\frac{c\mu}{2} + \frac{2(1-c)}{\mu} - 2c\right| \\ &\geq \left|\frac{1}{1-2c}\right|\left|2\sqrt{\frac{c\mu}{2}\frac{2(1-c)}{\mu}} - 2c\right| = \left|\frac{2\sqrt{c(1-c)} - c}{1-2c}\right|, \end{aligned}$$

with its minimum taking place at $\mu = \mu^*$ where

$$\frac{c\mu^*}{2} = \frac{2(1-c)}{\mu^*} \iff \mu^* = 2\sqrt{\frac{1-c}{c}} \tag{4.4.16}$$

Before claiming $\mu = \mu^*$ in (4.4.16) to be the choice for $\mu$, we verify its consistency with the previous selection rule (4.4.7) $1 < \mu < \frac{2}{c} - 2$ derived from leading terms only. This requires the following condition to be hold:

$$1 < 2\sqrt{\frac{1-c}{c}} < \frac{2}{c} - 2,$$

which is equivalent to

$$0 < c < \frac{1}{2}. \tag{4.4.17}$$

This condition is easily fulfilled by choosing the mesh size $h$ sufficiently small (this is because, as we have seen previously in (4.4.1), $c = h^2|\Omega_\infty| = h^2 \cdot \frac{L}{h} = hL$, it can be made arbitrarily small by decreasing mesh size $h$). Hence it is always possible to have formula (4.4.16) validated, which backs up the choice rule for $\mu^*$.

To sum up, we claim the following formula to be the parameter selection rule

$$\mu = \mu^* = 2\sqrt{\frac{1-c}{c}}, \tag{4.4.18}$$

with validation criterion $2 - (\mu + 2)c \backsimeq \mathcal{O}(1)$ and $0 < c < \frac{1}{2}$ fulfilled as stated in (4.4.14) and (4.4.17). The performance of this choice rule will be tested in the following section with numerical examples.

## 4.5 Implementation and Verification

### 4.5.1 Test on the Two-block Image

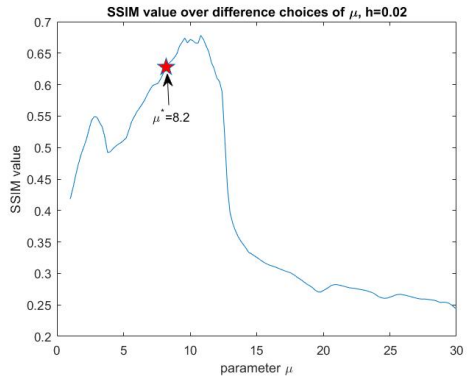
To test the performance of the heuristic formula (4.4.18) and justify its corresponding validation criterion numerically, we start from the simplest two-block example as

stated in (1.3.2). In this case, the total edge length is the perimeter of both rectangles, which sums up to $L = 2.8$. With $|\Omega_\infty| = \frac{L}{h}$ and $c = h^2|\Omega_\infty|$, the $c$ value for different mesh size $h$ may be calculated. Results in Table 4.1 and Fig 4.8 demonstrate the effectiveness of the choice rule $\mu = \mu^* = 2\sqrt{\frac{1-c}{c}}$ in (4.4.18) with different mesh size $h$, and scatter plots in Fig 4.9 justify this formula further:

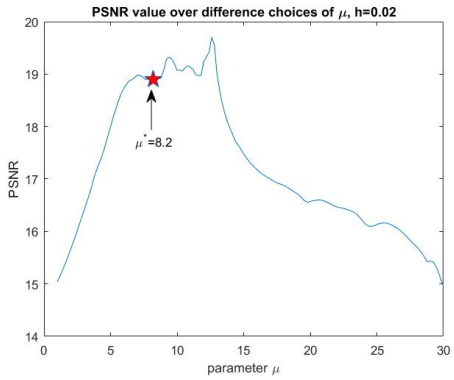
| $h$ | c | $\mu^*$ | SSIM | PSNR |
|---|---|---|---|---|
| $\frac{1}{50}$ | 0.056 | 8.2 | 0.63 | 18.90 |
| $\frac{1}{100}$ | 0.028 | 11.8 | 0.53 | 18.76 |
| $\frac{1}{200}$ | 0.014 | 16.8 | 0.53 | 18.69 |

Table 4.1: SSIM and PSNR index of the reconstructed image by the predicted value $\mu^*$ in the case of the two-block test image (1.3.2), with different mesh size $h$ for 10% Gaussian random noise. Corrupted image: SSIM=0.04, PSNR=13.16

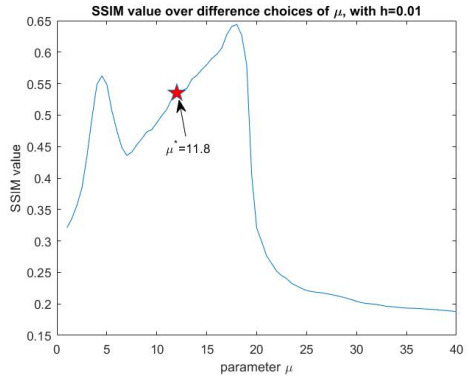
In addition to the above table, the following plots in Fig 4.8 demonstrate the ability of this parameter selection choice rule (4.4.18): for each given mesh size $h$, $\mu^*$ yields a solution which lies very close to the values that maximise SSIM or PSNR. Neither SSIM nor PSNR are exact measures of the quality of image reconstruction, and the difference between the values is well within the range acceptable for satisfactory reconstruction. The later plot Fig 4.10 shows that in the high noise level cases, the $\mu^*$ yielded by formula (4.4.18) leads to an image reconstruction that clearly outperforms the reconstruction obtained using a value of $\mu$ chosen by trial and error to maximise SSIM and PSNR.



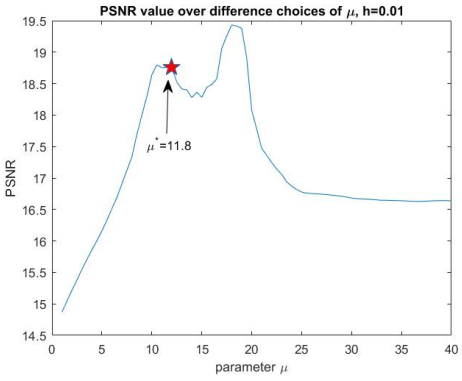

(a) SSIM, $h = \frac{1}{50}$



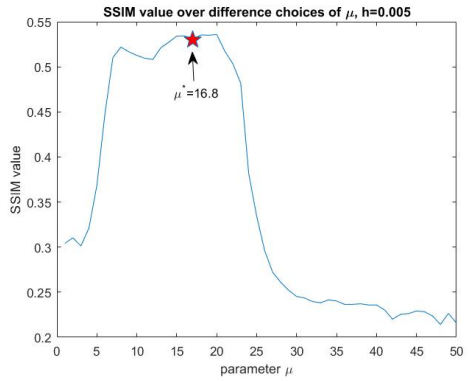

(b) PSNR, $h = \frac{1}{50}$



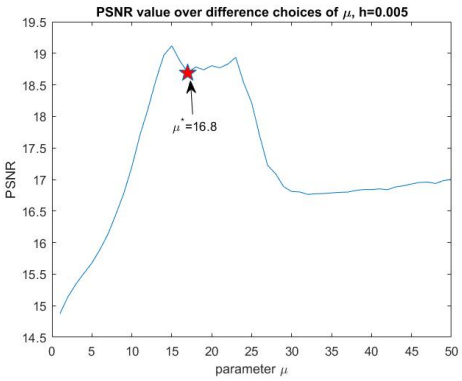

(c) SSIM, $h = \frac{1}{100}$




(d) PSNR, $h = \frac{1}{100}$

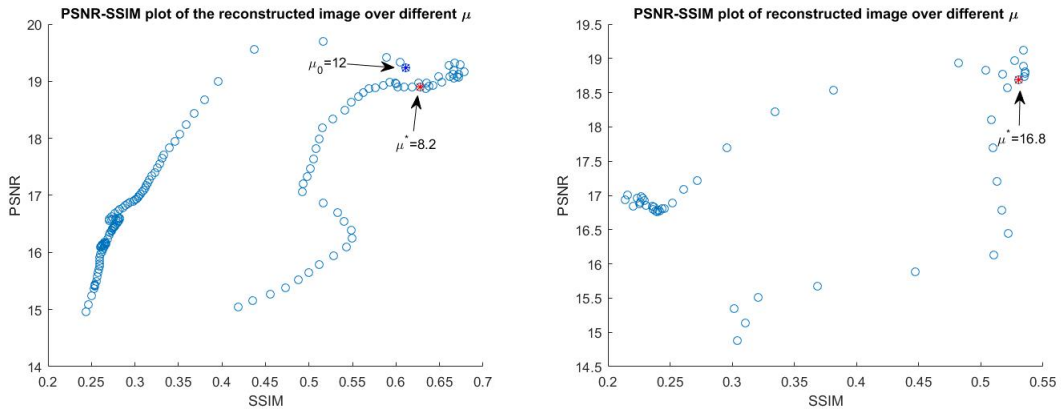



(e) SSIM, $h = \frac{1}{200}$




(f) PSNR, $h = \frac{1}{200}$

Figure 4.8: SSIM and PSNR index of the reconstructed images over different choices of $\mu$ in the case of 10% Gaussian random noise with different mesh size $h$

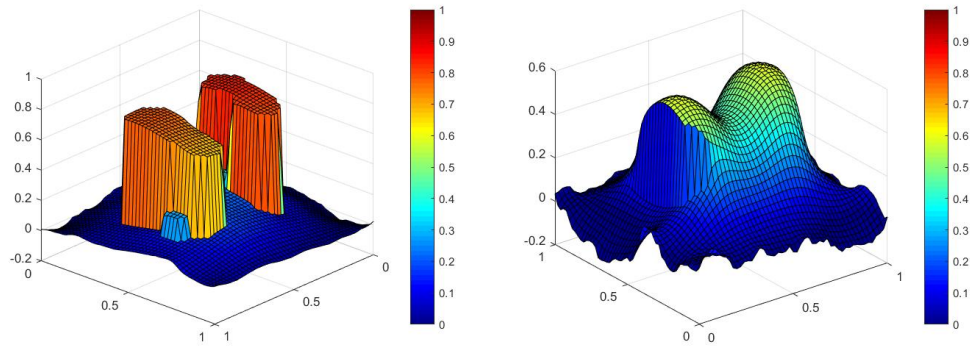



(a) $h = \frac{1}{50}$ (b) $h = \frac{1}{200}$

Figure 4.9: Scatter plot of PSNR-SSIM index of the reconstructed images over different choices of $\mu$ in the case of 10% Gaussian random noise with different mesh size $h$

**Robustness of the Formula**

It is worth pointing out that although my formula (4.4.18) does not depend on noise level $\epsilon$ explicitly, it is robust over different noise levels. To illustrate this point, consider the two choices of $\mu^* = 8.2$ and $\mu_0 = 12$ in Fig (4.9) (a), where both choices lead to reconstructions with very similar SSIM and PSNR index in the case of 10% Gaussian random noise. However, in Fig (4.10) with the case of 50% Gaussian random noise, $\mu^* = 8.2$ derived from my formula still recovers the major feature of the image, while $\mu_0 = 12$ results in a solution far away from the original one.



(a) reconstruction with $\mu^* = 8.2$ (b) $\mu_0 = 12$

Figure 4.10: Reconstructed images from $\mu^* = 8.2$ and $\mu_0 = 12$ in the case of 50% Gaussian random noise with mesh size $h = \frac{1}{50}$

### 4.5.2 Test on General Digital Images

In this section I will validate my formula $\mu = \mu^* = 2\sqrt{\frac{1-c}{c}}$ in (4.4.18) by justifying the natural fulfilment of condition

$$2 - (\mu + 2)c \backsimeq \mathcal{O}(1), \tag{4.5.1}$$

$$0 < c < \frac{1}{2} \tag{4.5.2}$$

using some more general examples. In the case of digital images, to calculate the value of $c$, which is essentially a ratio of "edge pixels" over all pixels of the image, we apply the Sobel filter with its automatic inbuilt threshold in the Matlab toolbox [R20a]. We use the following test images: (a) Shepp-Logan Phantom (256*256 pixels) which is purely a piecewise constant image; (b) Circle & Rectangle (200*300 pixels) which is a mixture of piecewise constant images and linear transition; and (c) Peppers (512*512 pixels) as a standard digital image. These figures and their Sobel filters are presented in Fig 4.11:

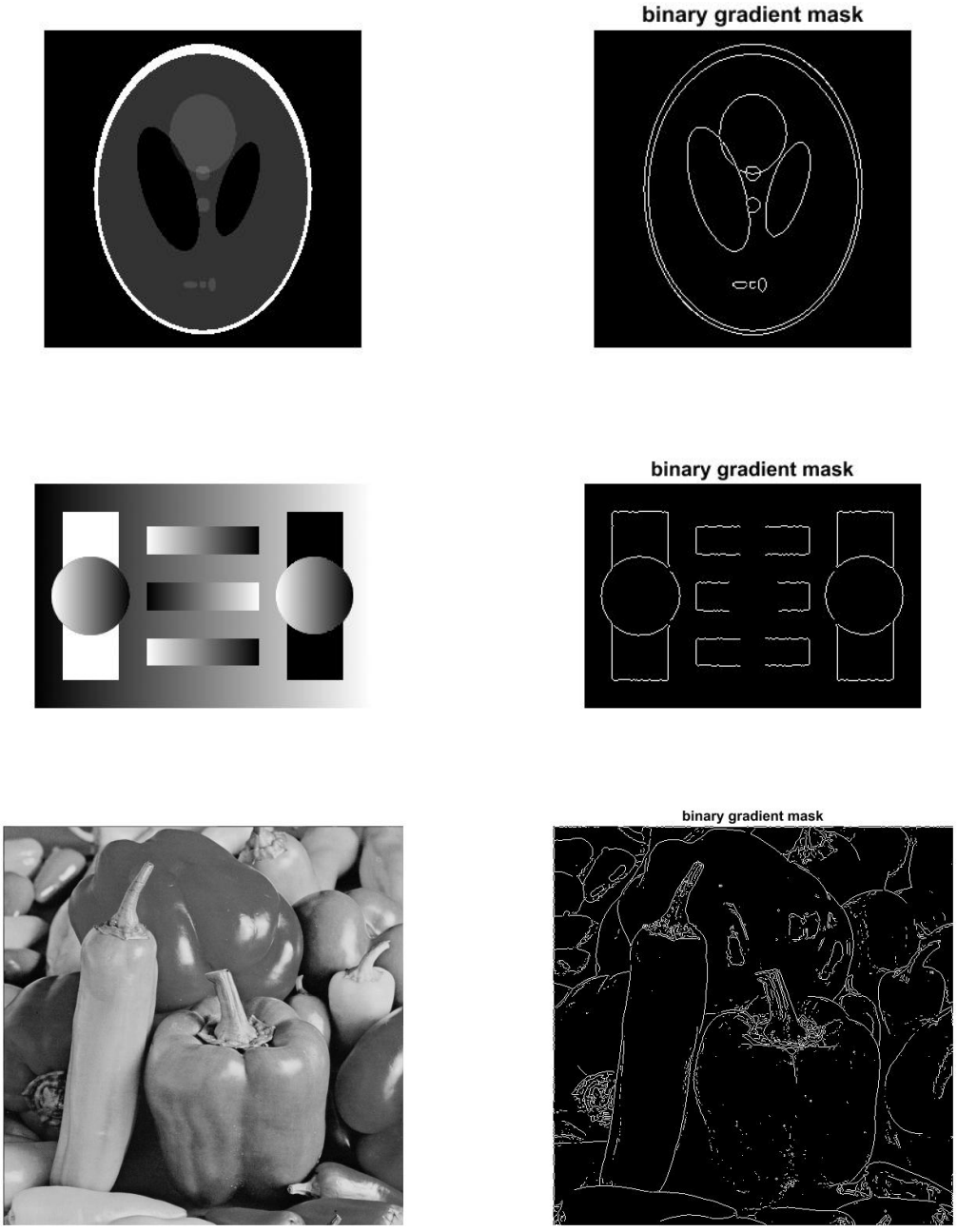



Figure 4.11: Exact digital images and their corresponding Sobel filter plot

It is worth pointing out that in most of the applications, when it is not possible to obtain $c$ from the exact image, we apply the Sobel filter to the corrupted image, and obtain an approximated $c_{cor}$ to calculate $\mu^*_{cor}$ accordingly. Fig 4.11 and Fig 4.12 show the Sobel filter of both the original and corrupted images, while Table 4.2 and Table 4.3 give the corresponding value of $c$ and $\mu^*$ in both exact and corrupted cases, as well as SSIM and PSNR index of their reconstruction respectively.

| Image | $c$ | $\mu^*$ | $2-(\mu+2)c$ | SSIM | PSNR |
|---|---|---|---|---|---|
| Shepp-Logan | 0.03 | 11.76 | 1.59 | 0.79 | 19.66 |
| Circle&Rectangle | 0.04 | 10.02 | 1.52 | 0.80 | 22.65 |
| Peppers | 0.05 | 8.82 | 1.46 | 0.69 | 24.95 |

Table 4.2: $c$ value and parameter $\mu^*$ calculated from exact digital images, and the SSIM and PSNR index of their reconstruction (with uniform 9*9 kernel and 10% Gaussian noise)

| Image | $c_{cor}$ | $\mu^*_{cor}$ | SSIM | PSNR |
|---|---|---|---|---|
| Shepp-Logan | 0.39 | 2.50 | 0.66 | 19.20 |
| Circle&Rectangle | 0.36 | 2.64 | 0.71 | 20.71 |
| Peppers | 0.26 | 3.37 | 0.71 | 25.90 |

Table 4.3: $c$ value calculated from corrupted digital images, and the SSIM and PSNR index of their reconstruction (with uniform 9*9 kernel and 10% Gaussian noise)

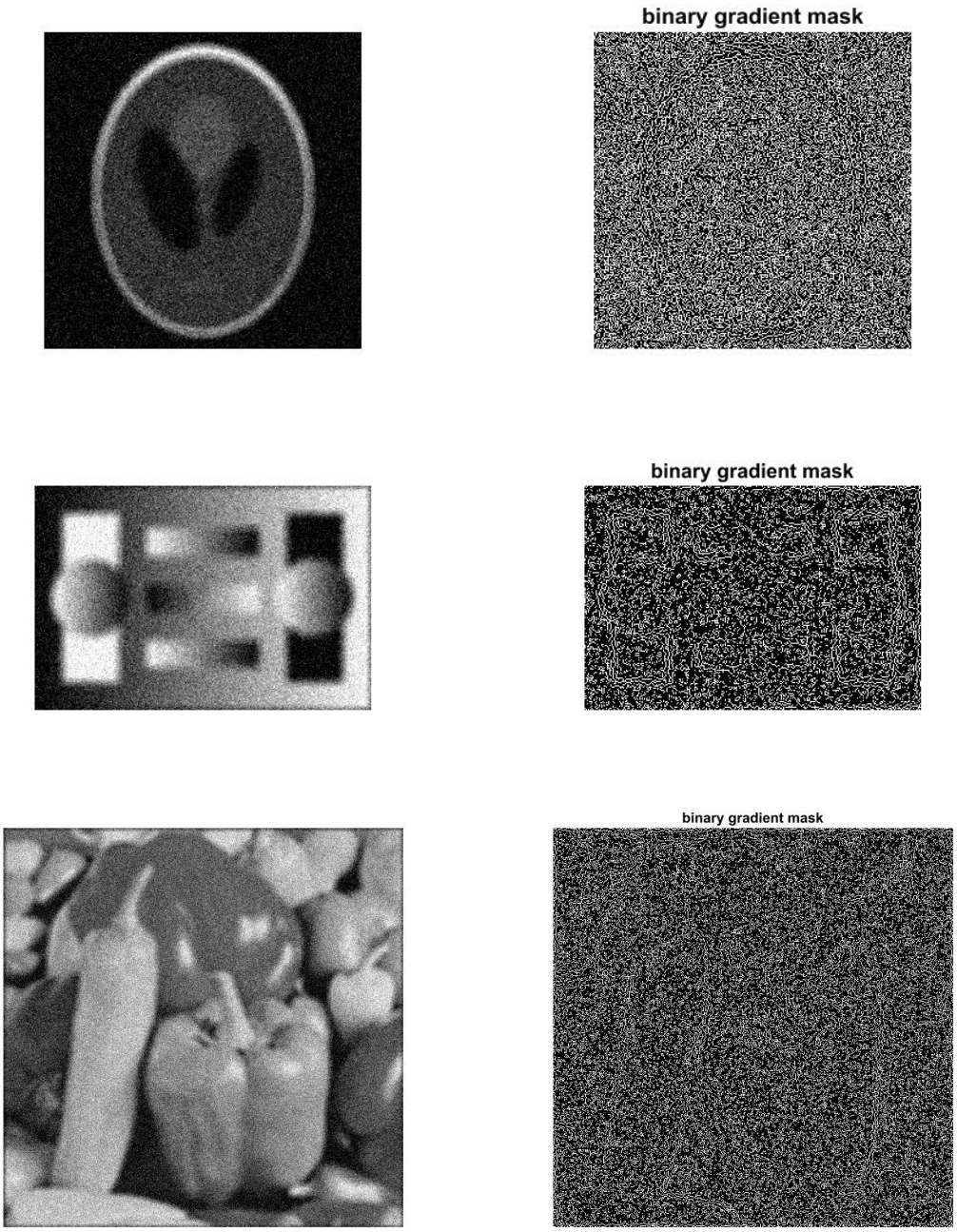
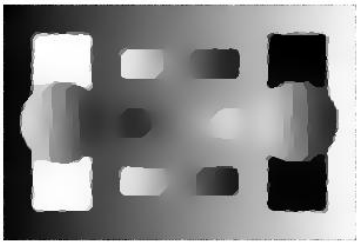


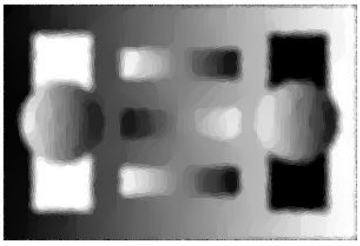

Figure 4.12: Corrupted digital images (with uniform 9*9 kernel and 10% Gaussian noise) and their corresponding Sobel filter plot

(a) $\mu^* = 11.76$

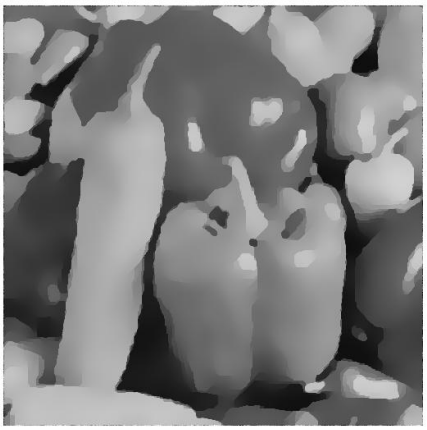
(b) $\mu^*_{cor} = 2.50$



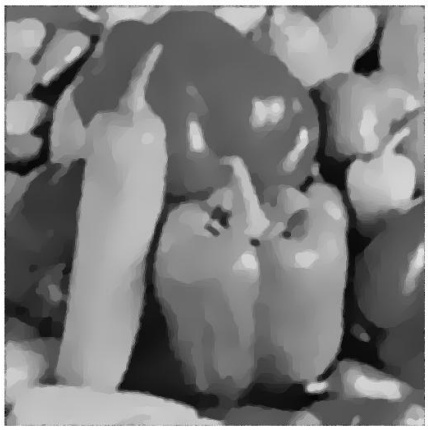
(c) $\mu^* = 10.02$



(d) $\mu^*_{cor} = 2.64$

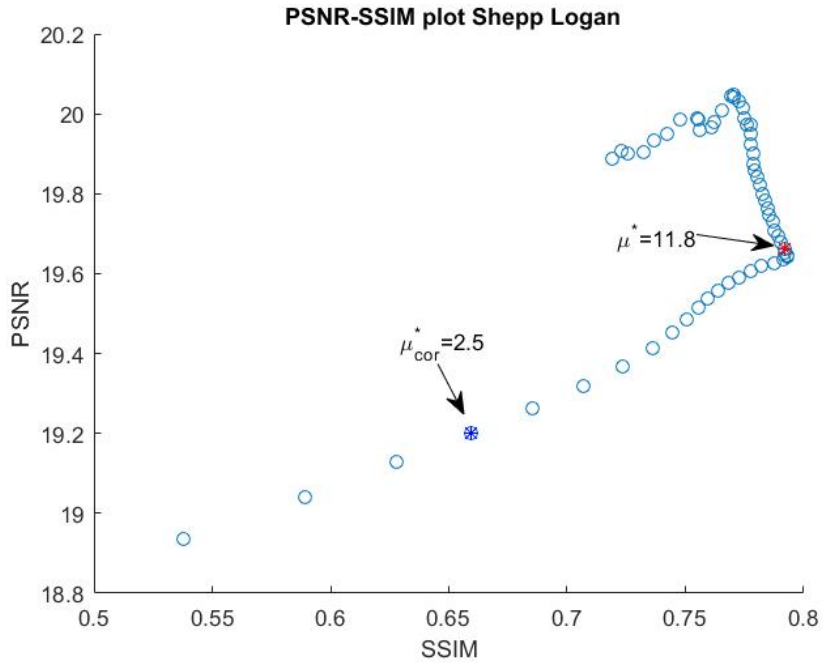


(e) $\mu^* = 8.82$

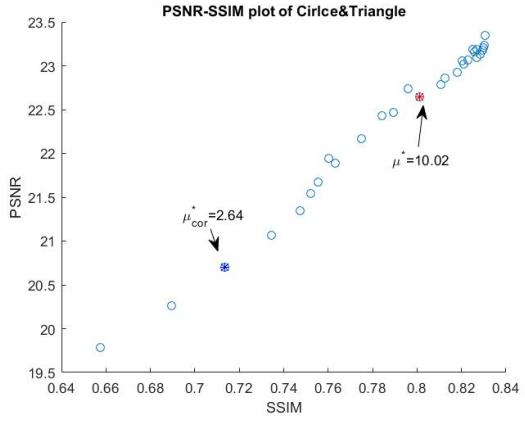


(f) $\mu^*_{cor} = 3.37$

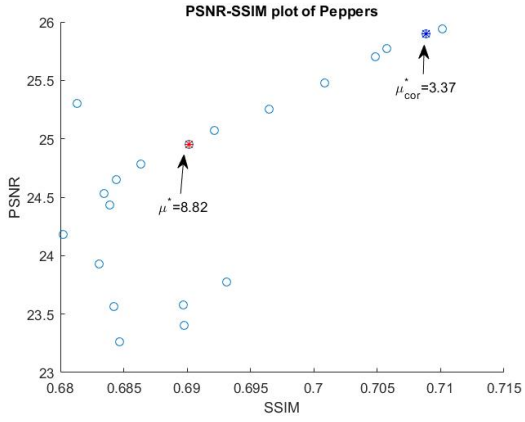
Figure 4.13: Reconstructions from $\mu^*$ calculated from exact image and $\mu^*_{cor}$ from corrupted image respectively (with uniform 9*9 kernel and 10% Gaussian noise)




(a) Shepp Logan Phantom




(b) Circle & Rectangle




(c) Peppers

Figure 4.14: Scatter plot of PSNR-SSIM index of the reconstructed images over different choices of $\mu$ for different digital images (with uniform 9*9 kernel and 10% Gaussian noise)

We can see from Table 4.2 that both condition (4.5.1) and (4.5.2) are satisfied

naturally, which justifies the derivation of formula (4.4.18). To further verify the reliability of this formula, we refer to Fig 4.14, which is a scatter plot of PSNR-SSIM index of the reconstructions by different $\mu$. Individual reconstructions by $\mu^*$ and $\mu^*_{cor}$ are presented in Fig 4.13, where it may be observed that

- For piecewise constant image such as the Shepp-Logan Phantom, or images that contain piecewise constant features to a large extent (e.g. Circle & Rectangle test image), although it is still possible to obtain satisfactory reconstructions from the estimated parameter $\mu^*_{cor}$ from the corrupted images, my choice rule $\mu^*$ calculated from the exact image leads to better reconstructions;

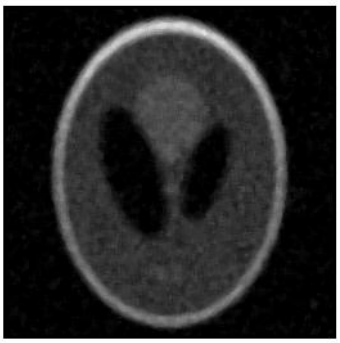
- For more complicated digital images such as illustrated by the example of "Peppers", this result is reverted: This is due to the stair-casing effect of the multiplicative regularisation, as demonstrated in Fig 4.13 (e) and (f).

To summarize, at relative low noise level (such as 10%), $\mu^*$ derived from the exact image performs better than $\mu^*_{cor}$ on piecewise constant images. However, for general digital images such as "Peppers", $\mu^*_{cor}$ outperforms $\mu^*$ due to the stair-casing effect. In the next few sections, I will discuss (a) how to estimate $\mu^*$ for piecewise constant images by smoothing the corrupted image first and (b) explanation of the stair-casing effect for general digital images.

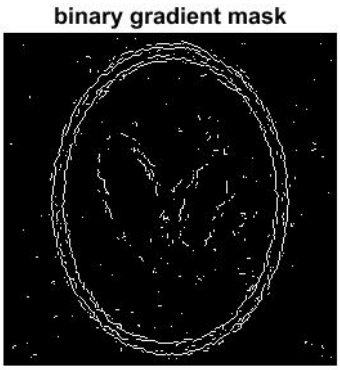
It is worth pointing out that at high noise level (such as 50%), $\mu^*$ derived from the exact image outperforms $\mu^*_{cor}$ for both piecewise constant and general digital images. This is demonstrated in Section 4.5.5, where a "two-step procedure" is provided to estimate $\mu^*$ for corrupted images. Various numerical examples of different nature show that the multiplicative model equipped with parameter selection rule (4.4.18) remains effective and robust for different noise distributions at high noise level.

### 4.5.3 Further Improvement for Piecewise Constant Images

As we have discussed in the previous section, using the value for the parameter $\mu^*_{cor}$ estimated directly from the corrupted image appears to limit the performance of multiplicative model, particularly in the case of piecewise constant images. To address this issue, in this section I present a possible way for further improvement: a preprocessing step to smooth the corrupted image first before applying the Sobel filter (or any other filter) for edge detection. Taking the Shepp-Logan Phantom (256*256 pixels) as an example, we firstly apply a $5 \times 5$ median filter [R20b] to the corrupted image in Fig 4.12. This gives rise to the smoothed image shown in Figure 4.15 (a). We then apply the Sobel filter with its automatic inbuilt threshold in the Matlab toolbox as discussed in [R20a] for edge detection, and obtain Figure 4.15 (b), whose quality is improved greatly from the previous Sobel filter plot obtained directly from corrupted image in Fig 4.12. Next, we estimate the values for $c$ and for further $\mu$ based on this Sobel filter plot. The corresponding reconstruction is presented in Fig 4.15(c). Both qualitative and quantitative results in Table 4.4 demonstrate that the result in Fig 4.15(c) is comparable with the reconstruction in Fig 4.13(a), which is obtained directly from $\mu^*$ based on the exact image.



(a) Smoothed image by median filter




(b) Sobel filter plot

(c) Reconstructed image

Figure 4.15: Smoothed image by median filter, its Sobel filter plot and corresponding reconstruction of Shepp-Logan Phantom (10 % Gaussian noise)

| Image | $c_{cor}$ | $\mu^*_{cor}$ | SSIM | PSNR |
|---|---|---|---|---|
| Shepp-Logan | 0.035 | 10.54 | 0.80 | 19.26 |

Table 4.4: $c$ value, $\mu_{cor}$ obtained from the smoothed Sobel filter in Fig 4.15(b), and the SSIM and PSNR index of the reconstruction in Fig 4.15(c)

It is worth mentioning that the choice of $5 \times 5$ median filter is determined by trial and error. This is feasible for relatively low noise level where the main image feature may be visualised. Different techniques such as the "two-step procedure" algorithm discussed in Section 4.5.5 will be applied for high noise level where the main features are not at all visible in the corrupted image.

In spite of its effectiveness on piecewise constant images as demonstrated using the Shepp-Logan Phantom, the power of this pre-processing technique is limited on more general digital images due to a "stair-casing effect". To illustrate this point, recall the example of "Peppers": results for this image presented in Table 4.2-4.3 and Fig 4.13 have demonstrated that the reconstruction obtained from $\mu_{cor}$ based on the corrupted image appears to be better both quantitatively and qualitatively. This means that the technique of pre-processing by smoothing the corrupted image before applying the Sobel filter for edge detection is not strictly necessary. We will explain the cause of this spurious effect and possible ways of mitigating it in the next section.

### 4.5.4 Understanding the Stair-casing Effect

As demonstrated in the numerical examples above, the multiplicative approach appears to produce visually more faithful reconstructions for piecewise constant images. On gray-level images, there appears to be spurious "stair-casing" effects in the reconstruction as illustrated in Fig 4.13 (c)-(f). To understand this undesirable phenomenon, let us temporarily consider the continuous gradient, instead of the finite difference in the discrete setting, and the model of a general digital image $u(x) \in L^1(\Omega)$. There are essentially three types of pixel points with different nature: (1) pixel $\mathbf{x}$ with a zero gradient $\nabla u(\mathbf{x}) = \mathbf{0}$; (2) pixel $\mathbf{x}$ which happens to be

a discontinuity and (3) pixel $\mathbf{x}$ with $\nabla u \neq \mathbf{0}$. According to our previous definition of "interior" and "edge" pixels, it is clear that pixels of type (1) and (2) should belong to "interior partition set" $\Omega_\infty^c$ and "edge partition set" $\Omega_\infty$, as they belong to asymptotic expansion regions $\{\mathbf{x}^c \big| \frac{|\nabla u(\mathbf{x}^c)|^2}{\delta_n^2} << 1\}$ and $\{\mathbf{x} \big| \frac{|\nabla u(\mathbf{x})|^2}{\delta_n^2} >> 1\}$ respectively (with a slight abuse of notation), regardless of the choice of steering parameter $\delta_n^2$. However, for the third type of pixels (for simplicity let us name them as "smooth pixels with non-zero gradient"), they belong to the asymptotic expansion region of $\{\mathbf{x} \big| \frac{|\nabla u(\mathbf{x})|^2}{\delta_n^2} \approx 1\}$, hence may be classified to either partition set $\Omega_\infty^c$ or $\Omega_\infty$ depending the choice of steering parameter $\delta_n^2$.

This ambiguity in recognition and separation of "edge pixels" and "interior pixels" leads to two types of issues of two aspects for general digital images. Firstly, it is difficult for the Sobel filter to classify these "smooth pixels with non-zero gradient" even for the original (exact) image, hence the value of tuning parameter $\mu^*$ cannot be chosen precisely. Secondly, in the multiplicative model, the "steering parameter" $\delta_n^2$ at each iteration acts as a threshold to differentiate "edge" and "interior" pixels during the restoration process. The sequence $\{\delta_n^2\}_{n\in\mathbb{N}}$ is then directed to its fixed point with a relatively small modulus. This means that the model is in practice pushing all pixel points to either "edge" ones which happens to be a discontinuity, or "interior" ones with zero gradient in $L^1(\Omega)$ space. In other words, the multiplicative model does not consider the third type of "smooth pixels with non-zero gradient" properly as one of its targets. As a result, for more complicated and general digital images which do include such pixel type, these pixels are reproduced as a combination of "interior" and "edge" pixels hence causing the stair-casing effect. Reconstructions of image "Rectangle & Circle" in Fig 4.3 (c)-(d) illustrate this phenomenon particularly: the stair-casing effect of the linear transition band appears to be heavier in the reconstruction using $\mu^*$ obtained from the exact image. Piecewise constant images are free from this effect due to the lack of "smooth pixels with non-zero gradient". This stair-casing effect explains why the multiplicative models works better on piecewise constant images than on gray-level images.

### 4.5.5 Robustness over High Noise Level and Different Distributions

In spite of the limitation of "stair-casing" effect for non piecewise constant digital images, the multiplicative model equipped with the parameter selection rule (4.4.18) remains a promising tool for image deblurring and denoising tool. Its main advantages is the robustness against high noise level and different noise distributions as discussed in previous Chapters. In this section, I will further demonstrate the effectiveness of multiplicative approach with parameter selection rule (4.4.18) using more general digital images.

Due to high noise level, the pre-processing technique of "smoothing the corrupted image prior to applying median filter or equivalence for edge detection" is no longer feasible in this case. However, we will show numerically that it is possible to obtain satisfactory reconstruction following a two-step procedure as proposed below:

1. Estimate $c_{corr}$ and $\mu^*_{cor}$ value base on corrupted image, and obtain a reconstruction based on these values (1st reconstruction);
2. Estimate $c_{corr}$ and $\mu^*_{cor}$ value again based on the 1st reconstruction, and obtain a second reconstruction based on these values. This will be used as the final outcome.

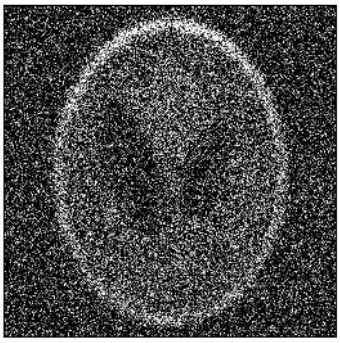
We test this "two-step procedure" using 50% Gaussian and K-distributed noise ($\alpha = 10$, asymmetric bell) on Shepp-Logan phantom and "Peppers", and obtain results as below. As shown in Table 4.5 and 4.6, and Figures 4.16 and 4.17 for the Shepp-Logan Phantom, and Table 4.7-4.8 and Figure 4.18-4.19 for Peppers, we find that for both images with different nature, the second reconstruction is significantly improved from the first reconstruction by employing a more accurate $\mu^*_{cor}$ value. The multiplicative model equipped with the "two-step procedure" appears to be robust against high noise level and different noise distribution.

| Image | $c_{cor}$ | $\mu^*_{cor}$ | SSIM | PSNR |
|---|---|---|---|---|
| Corrupted | N/A | N/A | 0.01 | 5.82 |
| 1st reconstruction | 0.27 | 3.27 | 0.25 | 16.27 |
| 2nd reconstruction | 0.08 | 6.77 | 0.61 | 17.43 |

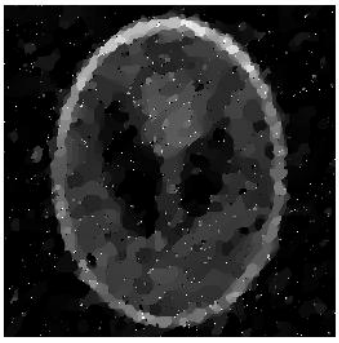
Table 4.5: Reconstruction of Shepp-Logan Phantom from image with 50% Gaussian noise using the two-step procedure

| Image | $c_{cor}$ | $\mu^*_{cor}$ | SSIM | PSNR |
|---|---|---|---|---|
| Corrupted | N/A | N/A | 0.01 | 5.77 |
| 1st reconstruction | 0.27 | 3.31 | 0.17 | 14.51 |
| 2nd reconstruction | 0.09 | 6.31 | 0.57 | 17.16 |

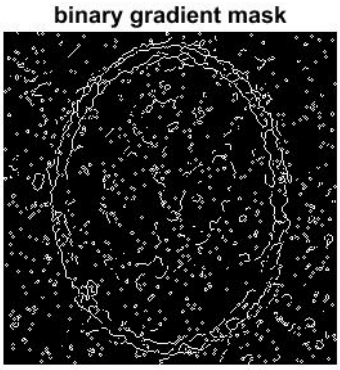
Table 4.6: Reconstruction of Shepp-Logan Phantom from image with 50% K-distributed noise ($\alpha = 10$, asymmetric bell) using the two-step procedure



(a) Corrupted image



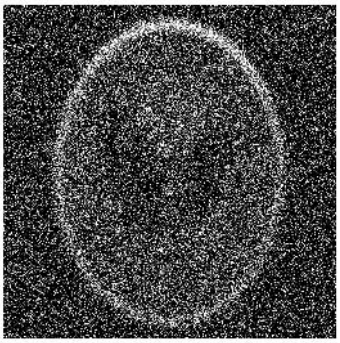
(b) 1st reconstruction



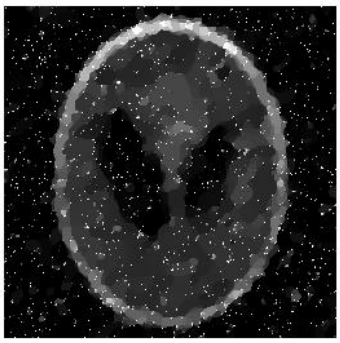

(c) Sobel filter plot obtained from 1st reconstruction

(d) 2nd reconstruction

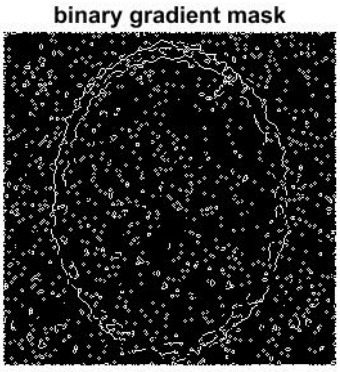
Figure 4.16: Reconstruction of Shepp-Logan Phantom from image with 50% Gaussian noise using the two-step procedure



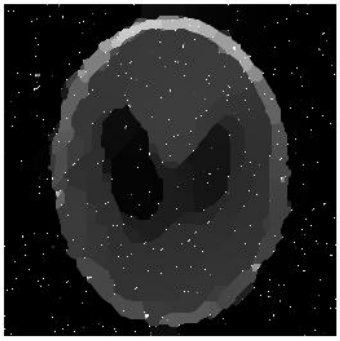
(a) Corrupted image



(b) 1st reconstruction




(c) Sobel filter plot obtained from 1st reconstruction



(d) 2nd reconstruction

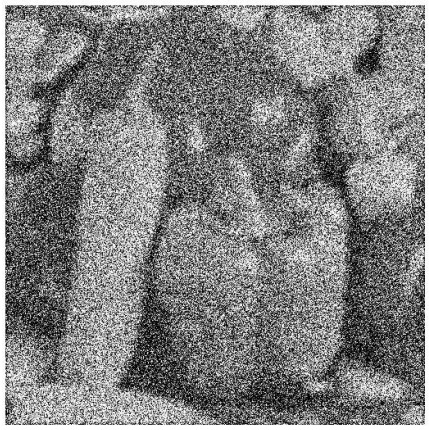
Figure 4.17: Reconstruction of Shepp-Logan Phantom from image with 50% K-distributed noise ($\alpha = 10$, asymmetric bell) using the two-step procedure

| Image | $c_{cor}$ | $\mu^*_{cor}$ | SSIM | PSNR |
|---|---|---|---|---|
| Corrupted | N/A | N/A | 0.01 | 5.98 |
| 1st reconstruction | 0.27 | 3.30 | 0.57 | 20.97 |
| 2nd reconstruction | 0.06 | 8.22 | 0.63 | 21.00 |

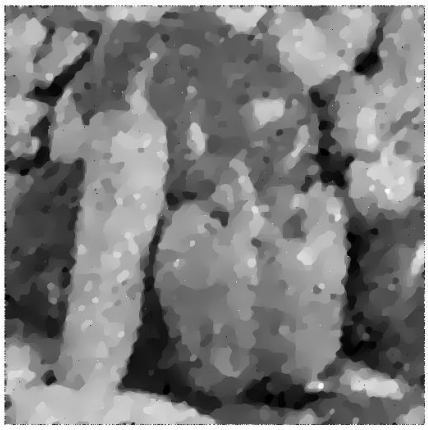
Table 4.7: Reconstruction of "Peppers" from image with 50% Gaussian noise using the two-step procedure

| Image | $c_{cor}$ | $\mu^*_{cor}$ | SSIM | PSNR |
|---|---|---|---|---|
| Corrupted | N/A | N/A | 0.01 | 5.97 |
| 1st reconstruction | 0.26 | 3.37 | 0.49 | 19.40 |
| 2nd reconstruction | 0.06 | 8.17 | 0.63 | 21.25 |

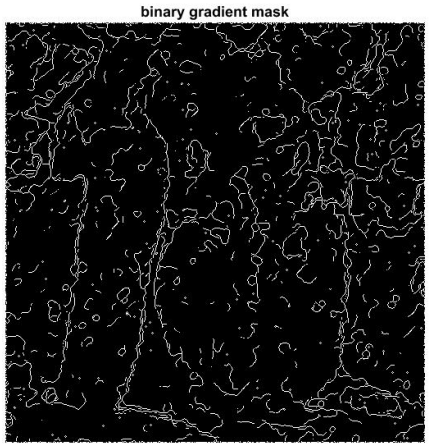
Table 4.8: Reconstruction of "Peppers" from image with 50% K-distributed noise ($\alpha = 10$, asymmetric bell) using the two-step procedure



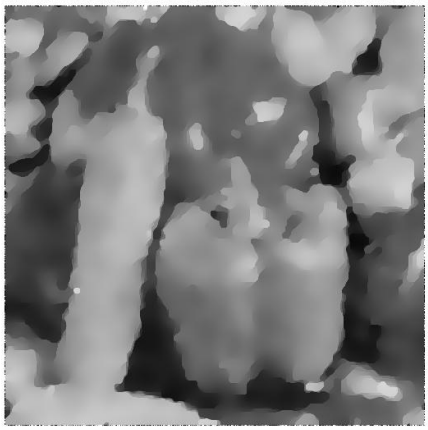
(a) Corrupted image



(b) 1st reconstruction

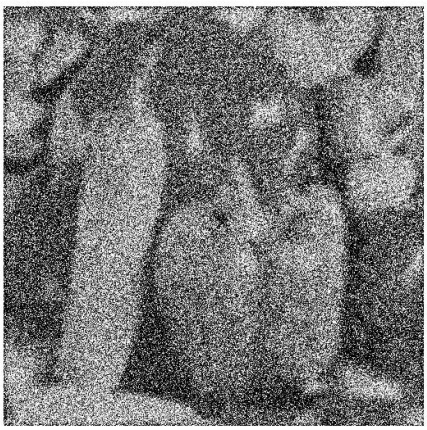



(c) Sobel filter plot obtained from 1st reconstruction

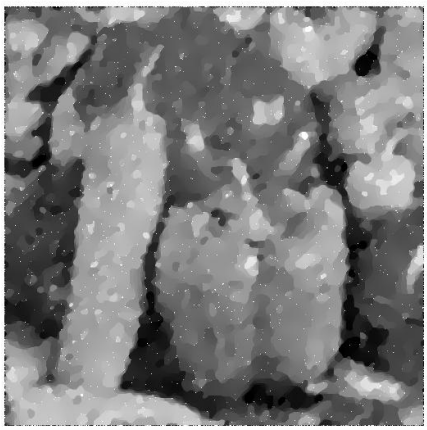


(d) 2nd reconstruction

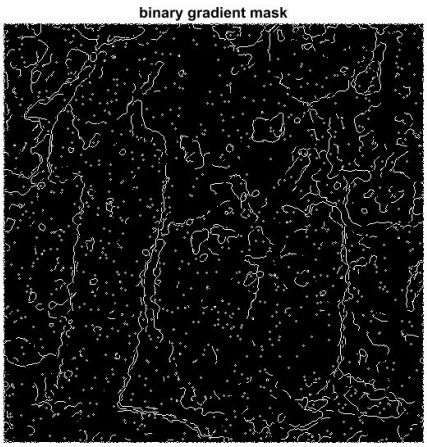
Figure 4.18: Reconstruction of "Peppers" from image with 50% Gaussian noise using the two-step procedure



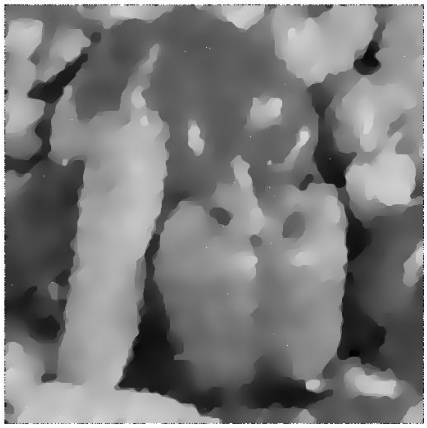
(a) Corrupted image



(b) 1st reconstruction




(c) Sobel filter plot obtained from 1st reconstruction



(d) 2nd reconstruction

Figure 4.19: Reconstruction of "Peppers" from image with 50% K-distributed noise ($\alpha = 10$, asymmetric bell) using the two-step procedure

### 4.5.6 Conclusion and Recommendation

In this chapter we proposed an explicit formula (4.4.18) which determines the $\mu$ value for obtaining best reconstruction in a straightforward manner. This parameter selection rule is demonstrated to be practical and robust by various numerical examples presented in this section. With the complement of this choice rule, Abubakar's multiplicative regularisation model [AvdBHB04] now reaches its full potential as a parameter-free method, with strength particularly on piecewise constant images. In practice, when the noise level is relatively low (10% or below), in the case of piecewise constant images, the value of this tuning parameter $\mu^*_{cor}$ is obtained by firstly smoothing the corrupted image by a median filter, and then applying formula (4.4.18) to obtain the estimation. This pre-processing step is not strictly necessary for more general digital images due to the limitation of "stair-casing" effect. For relatively high noise level (e.g. 50%) where smoothing the corrupted image prior to edge detection is not feasible, the "two-step procedure" as discussed in Section 4.5.4 may be applied. Numerical results has demonstrated this method to be effective and robust for general digital images of various noise distributions.

| | low level (e.g.10% noise) | high level (e.g.50% noise) |
|---|---|---|
| piecewise constant (e.g.Shepp-Logan) | $\mu = \mu^*$ obtained by smoothing the corrupted image first with a Median filter | $\mu = \mu^*$ obtained by the two-step procedure |
| general gray-level image (e.g."Pepper") | $\mu = \mu^*_{cor}$ due to the stair-casing effect | $\mu = \mu^*$ obtained by the two-step procedure |

Table 4.9: choice between $\mu^*$ and $\mu^*_{cor}$ for images with different nature and noise level

# Chapter 5

# Alternative Multiplicative Models

In this chapter and the next one, I derive a sequence of multiplicative regularisation models and will compare and relate them to the adaptive regularisation models by Abubakar *et al* [AvdBHB04] and Charbonnier *et al* [CBFAB97] as previously discussed. These multiplicative models are marriages between the Abubakar [AvdBHB04] and the Charbonnier [CBFAB97] model: the objective cost functional $C$ is constructed in the multiplicative form as a product of the fidelity and regularisation term, and is minimised by a modified alternating minimisation in terms of image intensity $u$, weight function $b$ and steering parameter $\delta$ (where applicable) iteratively. Note that the objective cost functional is nonlinear and non-convex in terms of image intensity $u$. We only require semi-convergence in the algorithm during the minimisation process of image intensity $u$.

The focus of this chapter is on my first and second formulation of such multiplicative models, which I call "Modified Multiplicative Regularisation" (MMR) and "Self-adaptive Scaling Parameter" (MSSP). In MMR, the scaling parameter $\delta_n \equiv \delta$ is a fixed constant (whose choice will be discussed later), while in MSSP, the parameter $\delta_n$ is self-adaptive in the algorithm. I will present both numerical and theoretical results for these two models, and demonstrate their effectiveness as well as provide insight into multiplicative models in general.

## 5.1 Model Formulation of MMR

Consider the following objective cost functional $\bar{C}(u)$ constructed as a product of the fidelity term and a multiplicative term which is expressed as a function of the image intensity $u_{i,j}$:

$$\bar{C}(u) = \{\sum_{(i,j)}[(\mathcal{K}u)_{i,j} - f^{\epsilon}_{i,j}]^2\} \cdot \{\sum_{(i,j)} \varphi\big(\sqrt{(D_x u)^2_{i,j} + (D_y u)^2_{i,j}}\big)\}, \tag{5.1.1}$$

with potential function

$$\varphi(t) = \varphi_{pot}(\frac{t}{\delta})$$

Here $t = t_{i,j} = \sqrt{(D_x u)^2_{i,j} + (D_y u)^2_{i,j}}$, $\varphi_{pot}$ takes one of the forms in Table 2.3 as discussed previously, and $\delta > 0$ is some fixed parameter which characterises the objective cost functional $\bar{C}(u)$. Using the potential function $\varphi(t)$ is advantageous in the setting of a multiplicative cost functional, even though $C(u)$ is of fourth order in $u$, since it enables us to derive some useful theoretical results. Also, as pointed out by [CBFAB97] and [GR92], this form of potential function is edge-preserving and therefore good for reconstructing piecewise constant images. Similarly to Charbonnier's model [CBFAB97], we use Theorem 2.5.1 and the existence of a strictly convex and decreasing function $\psi(\omega)$ to write $\varphi(t) = \omega t^2 + \psi(\omega)$ for every fixed $t$, therefore obtaining the objective cost functional $C(u, b)$ in the following form:

$$C(u, b^2) = \{\sum_{(i,j)}[(\mathcal{K}u)_{i,j} - f^{\epsilon}_{i,j}]^2\} \cdot \{\sum_{(i,j)} b^2_{i,j}[(D_x u)^2_{i,j} + (D_y u)^2_{i,j}] + \sum_{(i,j)} \psi(b^2_{i,j})\} \tag{5.1.2}$$

where $b^2 = b^2_{i,j} = \omega$ is the weight function. The two objective functionals are related via

$$\bar{C}(u) = \min_{b^2} C(u, b^2), \tag{5.1.3}$$

and it can be verified from Table 2.3 that $C(u, b)$ is convex in $b^2$. The form of the potential function $\psi$ may be derived from Theorem 2.5.1, and is given in Table 2.3. Given

$$\varphi(t) = \varphi_{pot}(\frac{t}{\delta}) = \inf_{\omega}(\frac{\omega}{\delta^2}t^2 + \psi_{pot}(\omega)) = \inf_{\omega^* = \frac{\omega}{\delta^2}}(\omega^* t^2 + \psi_{pot}(\delta^2\omega^*)),$$

we have:

$$\psi(b_{i,j}^2) = \psi_{pot}(\delta^2 b_{i,j}^2),$$

in eq (5.1.2) for each $(i, j)$.
Note that the objective cost functional $\bar{C}(u)$ in (5.1.1) is bounded below by 0, since $\varphi(t) \geq 0$, at any $t$. Direct application of the alternating minimisation algorithm (Algorithm 2.5.3) by Charbonnier *et al* [CBFAB97] to the multiplicative functional (5.1.1) would force its solution to a global minimum of $\bar{C}(u)$, which is achievable at either the value of $u$ where the fidelity term reaches 0, *i.e.*

$$\sum_{(i,j)}[(\mathcal{K}u)_{i,j} - f_{i,j}^{\epsilon}]^2 = 0, \tag{5.1.4}$$

or where the regularisation term reaches 0, *i.e.*

$$\sum_{(i,j)} \varphi\big(\sqrt{(D_x u)_{i,j}^2 + (D_y u)_{i,j}^2}\big) = \varphi_{pot}\Big(\frac{\sqrt{(D_x u)_{i,j}^2 + (D_y u)_{i,j}^2}}{\delta}\Big) = 0 \tag{5.1.5}$$

The former case of (5.1.4) leads to the solution of Moore Penrose generalised inverse for the noisy data $f^{\epsilon}$ (which always exists as we are in the discrete setting):

$$\hat{u} = u^{\dagger} = (\mathcal{K}^*\mathcal{K})^{-1}\mathcal{K}^* f^{\epsilon}, \tag{5.1.6}$$

while the latter case of (5.1.5) leads to the solution with $(D_x u)_{i,j} \equiv (D_y u)_{i,j} \equiv 0$ (recall properties of potential function $\varphi_{pot}$ in Chapter 2), *i.e.* any constant images

$$\hat{u}(\mathbf{x}) \equiv constant. \tag{5.1.7}$$

Neither of these two solutions (5.1.6) nor (5.1.7) is desirable from the image reconstruction point of view, as they correspond to over and under regularised solutions. To tackle this problem, we will adopt a "modified alternative minimisation algorithm" which consists of an "outer loop" for the weight function $b$, and an "inner loop" for the image intensity $u$. The model behaves like an iterative scheme for $u$, and exhibits semi-convergence in $u$, with the iteration index $n$ playing the role of the regularisation parameter. It is therefore equipped with a stopping criterion in the minimisation process of $u$. I will discuss in later sections how it is able to prevent the solution from converging to the wrong solution of constant images (5.1.7).

## 5.2 Modified Alternating Minimisation Algorithm

Following the discussion in previous section, we develop here a two-step "modified alternating minimisation algorithm" for MMR, which is similar to the previous Algorithm 3.2.1 by Abubakar *et al* [AvdBHB04] but has the following improvement and differences: (1) employing potential function $\varphi_{pot}$ in the algorithm; (2) separation of the algorithm into an inner and an outer loop for variables $b$ and $u$ respectively; (3) choosing step size $\alpha_{n+1}^{m+1}$ in such a way that the objective cost functional is guaranteed to decrease monotonically.

To explain the notation, we use subscript $n$ to index the outer loop, and superscript $m$ for the inner one. $M_n$ denotes the total number of inner loop iterations conducted at the $n$th outer loop, with the complementary variable $b = b_n$ fixed, *i.e.*,

$$u_{n+1}^{0} = u_n^{M_n}.$$

We will use $M_{outer}$ and $M_{inner}$ to denote the maximum number of iteration conducted in the outer and inner loop respectively. In practice, we choose $M_{inner}$ to be approximately $10-30$, and $M_{outer}$ to be $200-300$. Tolerance *tol* in the termination criterion is chosen to be $10^{-7}$.

**Algorithm 5.2.1.** ***Modified alternating minimisation algorithm***
*Initialization:* $u_0^{M_0} = f^\epsilon$, *choose* $M_{inner}$ *and* $M_{outer}$*;*
*For* $0 \leq n \leq M_{outer}$, *Repeat*
***Outer loop:*** *Iterate for* $b$ *on the subscript* $n$

$$(b_{n+1}^2)_{i,j} = \frac{\varphi'(\sqrt{(D_x u_n^{M_n})_{i,j}^2 + (D_y u_n^{M_n})_{i,j}^2})}{2(D_x u_n^{M_n})_{i,j}^2 + (D_y u_n^{M_n})_{i,j}^2}, \ \forall\ i,j = 1,2,...N. \tag{5.2.1}$$

***Inner loop:*** *Fix* $b = b_{n+1}$ *and iterate for* $u$ *on the superscript* $m$
*Initialization:* $u_{n+1}^0 = u_n^{M_n}$.
*If* $1 \leq m \leq M_{inner}$, *then*
*Update image intensity* $u_{n+1}^{m+1}$*:*

$$u_{n+1}^{m+1} = u_{n+1}^m + \alpha_{n+1}^{m+1} v_{n+1}^{m+1}, \ m \geq 0,$$

*where search direction* $v_{n+1}^{m+1}$ *is defined via*

$$\begin{aligned} \boldsymbol{v}_{n+1}^0 &= -\boldsymbol{g}_{n+1}^0 \\ \boldsymbol{v}_{n+1}^{m+1} &= -\boldsymbol{g}_{n+1}^{m+1} + \frac{(\boldsymbol{g}_{n+1}^{m+1})^T \boldsymbol{g}_{n+1}^{m+1}}{(\boldsymbol{g}_{n+1}^m)^T \boldsymbol{g}_{n+1}^m} \boldsymbol{v}_{n+1}^m, \ m \geq 1 \end{aligned} \tag{5.2.2}$$

*with gradient* $\boldsymbol{g}_{n+1}^m$ *being*

$$\boldsymbol{g}_{n+1}^m = \boldsymbol{K}^T(\boldsymbol{K}\boldsymbol{u}_{n+1}^m - \boldsymbol{f}^\epsilon) + \frac{F(u_{n+1}^m)}{F_{n+1}^{multi}(u_{n+1}^m, b_{n+1}^2)} \boldsymbol{D}^T \boldsymbol{B}_{n+1} \boldsymbol{D}\boldsymbol{u}_{n+1}^m, \ m \geq 0 \tag{5.2.3}$$

*and step size* $\alpha_{n+1}^{m+1}$ *chosen to be the smallest positive real root of the cubic polynomial*

*in $\alpha$:*

$$\frac{\partial}{\partial\alpha}C(u_{n+1}^m + \alpha v_{n+1}^{m+1}, b_{n+1}^2)\big|_{\alpha=\alpha_{n+1}^{m+1}} = 0 \tag{5.2.4}$$

***Termination criterion for inner loop:*** *If*

$$||u_{n+1}^{m+1} - u_{n+1}^m||_2 < tol,\ for\ some\ 1 \le m \le M_{inner},$$

*or iteration index m reaches $M_{inner}$, then stop and go to outer loop.*
***Termination criterion for outer loop:*** *If*

$$||u_{n+1}^{M_{n+1}} - u_n^{M_n}||_2 < tol,\ for\ some\ 1 \le n \le M_{outer},$$

*or iteration index n reaches $M_{outer}$, then stop the algorithm.*

Here we adopt similar notation to Algorithm 3.2.1 for simplicity of calculation. $\mathbf{u}_{n+1}^m$ and $\mathbf{f}^\epsilon$ are reshaped 2D image intensities $u_{n+1}^m$ and $f^\epsilon$ from $N \times N$ matrices to 1D arrays with dimension $1 \times N^2$. In (5.2.3) $\mathbf{K}$, $\mathbf{D}$ and $\mathbf{B}_{n+1}$ are all $N^2 \times N^2$ matrices representing the convolution operator $\mathcal{K}$, the finite difference operator $D_x$, $D_y$ and the weight function $(b_{n+1}^{m+1})^2$ correspondingly in the re-shaped setting. The gradients $\mathbf{g}_{n+1}^m$, $\mathbf{g}_{n+1}^{m+1}$ and the search directions $\mathbf{v}_{n+1}^m$, $\mathbf{v}_{n+1}^{m+1}$ are also $1 \times N^2$ arrays in (3.2.5). Specifically, the matrix $\mathbf{B}_{n+1}$ takes the form of

$$\mathbf{B}_{n+1} = diag\{(b_{n+1}^2)_k,\ k = 1, 2, ...N^2\},$$

where the entries $(b_{n+1}^2)_k$ corresponds to $(b_{n+1}^2)_{i,j},\ 1 \le i, j \le N$.
The multiplicative term in (5.2.3) takes the following form:

$$F_{n+1}^{multi}(u_{n+1}^m, b_{n+1}^2) = \sum_{(i,j)}(b_{n+1}^2)_{i,j}[(D_x u_{n+1}^m)_{i,j}^2 + (D_y u_{n+1}^m)_{i,j}^2] + \sum_{(i,j)}\psi\big((b_{n+1}^2)_{i,j}\big)$$

**Monotonic decrease in objective cost functional** $C(u, b)$
In spite of their similarities, one advantage of Algorithm 5.2.1 for MMR over Abubakar's

Algorithm 3.2.1 is the theoretically robustness, which is reflected over the choice of step size $\alpha_{n+1}^m$ in the inner loop. Recall from previous Section 3.2.4, Algorithm 3.2.1 simply extended to the nosy case the convexity results obtained in the noiseless case, hence determined step size $\alpha_{n+1}$ in the noisy case without explicit theoretical justification. This is not the case for MMR. Algorithm 5.2.1 on the other hand chooses step size $\alpha_{n+1}^m$ in order to enforce that the objective cost functional $C(u, b)$ decreases monotonically. This is an important result and serves as the foundation for later convergence analysis. Note that in the outer loop, the update rule for the weight function $b_{n+1}^2$ in (5.2.1) is derived using Theorem 2.5.1, which states that the objective cost functional $C(u, b)$ has a unique minimum in $b^2$ achievable at (2.5.4). Monotonic decrease of $C(u, b)$ is hence guaranteed in every outer loop, which leaves us with the inner loop case for further discussion.

**Proposition 5.2.2.** *If $\alpha_{n+1}^{m+1}$ is chosen as the smallest positive real root of the cubic polynomial (5.2.4) in $\alpha$, then*

$$C(u_{n+1}^{m+1}, b_{n+1}^2) = C(u_{n+1}^m + \alpha_{n+1}^{m+1} v_{n+1}^{m+1}, b_{n+1}^2) \leq C(u_{n+1}^m, b_{n+1}^2)$$

This is sufficient to conclude the monotonic decrease of $C(u, b)$ in the inner loop of Algorithm 5.2.1. Before proving Proposition 5.2.2, let us consider the following Lemma on gradient descent of the conjugate gradient method first:

**Lemma 5.2.3.** *For gradient $g_{n+1}^m$ and search direction $v_{n+1}^m$ defined as in Algorithm 5.2.1,*

$$\left\langle g_{n+1}^{m+1}, v_{n+1}^{m+1} \right\rangle_2 = -||g_{n+1}^{m+1}||_2^2 < 0 \tag{5.2.5}$$

*Proof.* We start from the definition of the search direction $v_{n+1}^{m+1}$ as given in (5.2.2). For $m = 0$, the conjugate gradient method becomes a steepest descent one, and it is straightforward to show

$$\left\langle g_{n+1}^0, v_{n+1}^0 \right\rangle_2 = -||g_{n+1}^0||_2^2 < 0.$$

For $m \geq 1$, if

$$\left\langle g_{n+1}^{m+1}, v_{n+1}^{m} \right\rangle_2 = 0. \tag{5.2.6}$$

holds, then the proof of the lemma will be complete. We show next that (5.2.6) holds. Consider the $m$th iteration of the inner loop:

$$u_{n+1}^{m} = u_{n+1}^{m-1} + \alpha_{n+1}^{m} v_{n+1}^{m}, \tag{5.2.7}$$

where step size $\alpha_{n+1}^{m}$ is calculated as

$$\frac{\partial}{\partial \alpha} C(u_{n+1}^{m-1} + \alpha v_{n+1}^{m}, b_{n+1}^{2})|_{\alpha=\alpha_{n+1}^{m}} = 0$$

By definition of the derivative, this may also be expressed as

$$\lim_{h \to 0} \frac{1}{h} \{C[u_{n+1}^{m-1} + (\alpha_{n+1}^{m} + h) v_{n+1}^{m}, b_{n+1}^{2}] - C[u_{n+1}^{m-1} + \alpha_{n+1}^{m} v_{n+1}^{m}, b_{n+1}^{2}]\} = 0,$$

and re-written as

$$\lim_{h \to 0} \frac{1}{h} \{C[(u_{n+1}^{m-1} + \alpha_{n+1}^{m} v_{n+1}^{m}) + h v_{n+1}^{m}, b_{n+1}^{2}] - C[u_{n+1}^{m-1} + \alpha_{n+1}^{m} v_{n+1}^{m}, b_{n+1}^{2}]\} = 0, \tag{5.2.8}$$

The definition of the directional derivative gives:

$$\lim_{h \to 0} \frac{1}{h} \{C(u + h\omega, b^2) - C(u, b^2)\} = \left\langle \nabla_u C(u, b^2), \omega \right\rangle_2, \ \forall \omega. \tag{5.2.9}$$

Take

$$u = u_{n+1}^{m-1} + \alpha_{n+1}^{m} v_{n+1}^{m}$$
$$\omega = v_{n+1}^{m},$$

it follows from (5.2.8) and (5.2.9) that

$$\left\langle \nabla_u C[u_{n+1}^{m-1} + \alpha_{n+1}^{m} v_{n+1}^{m}, b_{n+1}^{2}], v_{n+1}^{m} \right\rangle_2 = 0, \tag{5.2.10}$$

Recall the definition of gradient $g_{n+1}^{m+1}$:

$$g_{n+1}^{m+1} = \nabla_u C[u_{n+1}^m, b_{n+1}^2] = \nabla_u C[u_{n+1}^{m-1} + \alpha_{n+1}^m v_{n+1}^m, b_{n+1}^2]. \tag{5.2.11}$$

(5.2.6) may be constructed by substituting (5.2.11) into (5.2.10), which completes the proof.

□

Lemma 5.2.3 shows that the search direction $v_{n+1}^{m+1}$ in Algorithm 5.2.1 guarantees gradient descent of the cost functional $C(u_{n+1}^m + \alpha v_{n+1}^{m+1}, b_{n+1}^2)$ at every iteration of the inner loop. Mathematically this means there exists some $\alpha > 0$ such that

$$C(u_{n+1}^m + \alpha v_{n+1}^{m+1}, b_{n+1}^2) \leq C(u_{n+1}^m, b_{n+1}^2). \tag{5.2.12}$$

To complete the proof for Proposition 5.2.2, we still need to show that the smallest positive real root of the cubic polynomial (5.2.4) in $\alpha$ indeed satisfies (5.2.12). Since the 4th order polynomial $C(u_{n+1}^m + \alpha v_{n+1}^{m+1}, b_{n+1}^2)$ in $\alpha$ is bounded below by 0, and because of the choice (5.2.2) for the search direction, all possibilities for function $C(u_{n+1}^m + \alpha v_{n+1}^{m+1}, b_{n+1}^2)$ in terms of $\alpha$ reduce to the three cases shown in Figure (5.1)-(5.2) below. The starting position $u_{n+1}^m$ corresponds to $\alpha = 0$ on these graphs. We have verified in every case that when starting from $\alpha = 0$ and moving along a gradient descent direction as arrow pointed, the first critical point we will reach indeed guarantees(5.2.12) hence ensures the monotonic decrease of $\{C(u_{n+1}^m, b_{n+1}^2)\}_{m\in\mathbb{N}}$ for each fixed outer loop $n + 1$.

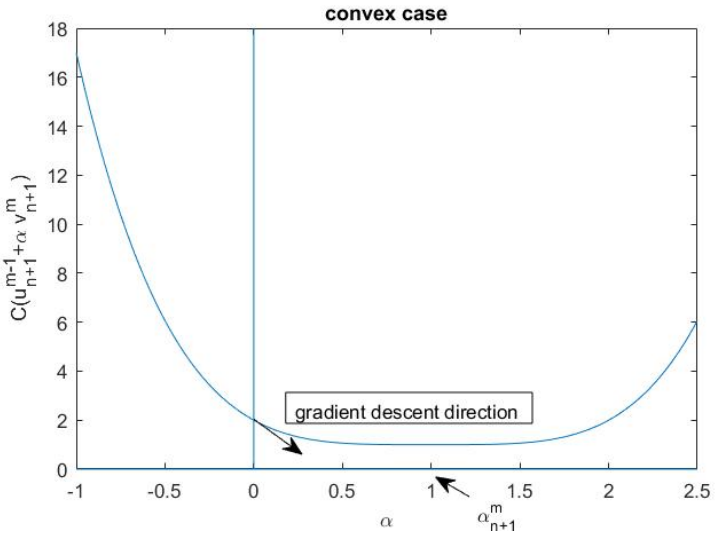
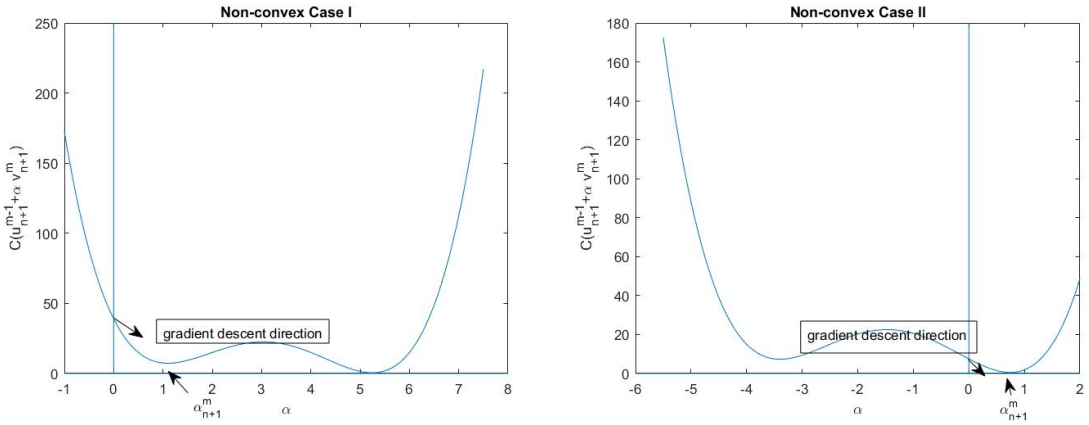



Figure 5.1: Convex case




(a) non-convex case I: (b) non-convex case II

The search method in the inner loop of the iterative scheme used here is an example of the method of **feasible directions** classified by Zoutendijk [Zou70], (see Chapter 2.3.4 in the Mathematical Preliminaries of this thesis). In fact, it may be replaced by or generalised to any other search method which guarantees gradient descent in the search direction, *i.e.* $\left\langle g_{n+1}^{m+1}, v_{n+1}^{m+1} \right\rangle < 0$ in Lemma 5.2.3. Such numerical algorithms

choose the step size $\alpha_{n+1}^m$ in each iteration of the inner loop so that

$$\begin{aligned} &C(u_{n+1}^{m+1}, b_{n+1}^2) = C(u_{n+1}^m + \alpha_{n+1}^{m+1} v_{n+1}^{m+1}, b_{n+1}^2) \\ &\leq C(u_{n+1}^m + \alpha v_{n+1}^{m+1}, b_{n+1}^2)|_{\alpha=0} = C(u_{n+1}^m, b_{n+1}^2), \end{aligned} \tag{5.2.13}$$

**Avoidance of constant image reconstruction**

In this section we will demonstrate the ability of Algorithm 5.2.1 to prevent its solution to the constant image reconstruction (5.1.7). By substituting both the Moore Penrose generalised inverse with noisy data $u^\dagger = (\mathcal{K}^*\mathcal{K})^{-1}\mathcal{K}^* f^\epsilon$ in (5.1.6) and the constant image $u(\mathbf{x}) \equiv constant$ in (5.1.7) into gradient expression (5.2.3), it may be checked that zero gradient $g_{n+1}^m = 0$ only occurs at the Moore Penrose generalised inverse of $u^\dagger = (\mathcal{K}^*\mathcal{K})^{-1}\mathcal{K}^* f^\epsilon$. This demonstrates its ability to avoid the convergence to the wrong solution of constant images. In the cases where Algorithm 5.2.1 directs the solution to the Moore Penrose generalised inverse of $u^\dagger = (\mathcal{K}^*\mathcal{K})^{-1}\mathcal{K}^* f^\epsilon$, the maximum iteration numbers $M_{inner}$ and $M_{outer}$ ensure that the algorithm terminates in time, and play the role of a regularisation parameter in MMR.

**Summary of MMR with different potential functions**

For completeness and ease of reference, we give below the explicit form of the objective cost functional $\bar{C}(u)$ in (5.1.1), $C(u, b)$ in (5.1.2) and of the recurrence relation for the weight function $(b_{n+1}^2)_{i,j}$ in (5.2.1) in MMR with some common choices of potential functions (see Table 2.3 for more details):

HL type:

$$\begin{aligned} \bar{C}_{HL}(u) &= \{\sum_{(i,j)}[(\mathcal{K}u)_{i,j} - f_{i,j}^\epsilon]^2\} \cdot \{\sum_{(i,j)} \log(1 + \frac{(D_x u)_{i,j}^2 + (D_y u)_{i,j}^2}{\delta^2})\} \\ C_{HL}(u, b^2) &= \{\sum_{(i,j)}[(\mathcal{K}u)_{i,j} - f_{i,j}^\epsilon]^2\} \cdot \{\sum_{(i,j)} b_{i,j}^2[(D_x u)_{i,j}^2 + (D_y u)_{i,j}^2] + \sum_{(i,j)}[\delta^2 b_{i,j}^2 - \log(\delta^2 b_{i,j}^2) - 1]\} \\ (b_{n+1}^2)_{i,j} &= \frac{1}{(D_x u_n^{M_n})_{i,j}^2 + (D_y u_n^{M_n})_{i,j}^2 + \delta^2} \ \forall\ i, j = 1, 2, ...N \end{aligned}$$

GM type:

$$\begin{aligned}
\bar{C}_{GM}(u) &= \{\sum_{(i,j)}[(\mathcal{K}u)_{i,j} - f^{\epsilon}_{i,j}]^2\} \cdot \left\{ \sum_{(i,j)} \frac{(D_x u)^2_{i,j} + (D_y u)^2_{i,j}}{\delta^2 + (D_x u)^2_{i,j} + (D_y u)^2_{i,j}} \right\}, \\
C_{GM}(u, b^2) &= \{\sum_{(i,j)}[(\mathcal{K}u)_{i,j} - f^{\epsilon}_{i,j}]^2\} \cdot \{\sum_{(i,j)} b^2_{i,j}[(D_x u)^2_{i,j} + (D_y u)^2_{i,j}] + \sum_{(i,j)}[\delta^2 b^2_{i,j} - 2(\delta b_{i,j}) + 1]\}, \\
(b^2_{n+1})_{i,j} &= \frac{\delta^2}{[(D_x u^{M_n}_n)^2_{i,j} + (D_y u^{M_n}_n)^2_{i,j} + \delta^2]^2} \ \forall\ i, j = 1, 2, ...N.
\end{aligned}$$

HS type:

$$\begin{aligned}
\bar{C}_{HS}(u) &= \{\sum_{(i,j)}[(\mathcal{K}u)_{i,j} - f^{\epsilon}_{i,j}]^2\} \cdot \left\{ \sum_{(i,j)} 2\sqrt{1 + \frac{(D_x u)^2_{i,j} + (D_y u)^2_{i,j}}{\delta^2}} - 2 \right\}, \\
C_{HS}(u, b^2) &= \{\sum_{(i,j)}[(\mathcal{K}u)_{i,j} - f^{\epsilon}_{i,j}]^2\} \cdot \{\sum_{(i,j)} b^2_{i,j}[(D_x u)^2_{i,j} + (D_y u)^2_{i,j}] + \sum_{(i,j)}[\delta^2 b^2_{i,j} + \frac{1}{\delta^2 b^2_{i,j}} - 2]\}, \\
(b^4_{n+1})_{i,j} &= \frac{1}{\delta^2[(D_x u^{M_n}_n)^2_{i,j} + (D_y u^{M_n}_n)^2_{i,j} + \delta^2]} \ \forall\ i, j = 1, 2, ...N.
\end{aligned}$$

**Additive Counterpart of MMR**

The multiplicative model MMR with objective cost functional which has the general form

$$C^{MMR}(u, b^2) = \{\sum_{(i,j)}[(\mathcal{K}u)_{i,j} - f^{\epsilon}_{i,j}]^2\} \cdot \{\sum_{(i,j)} b^2_{i,j}[(D_x u)^2_{i,j} + (D_y u)^2_{i,j}] + \sum_{(i,j)} \psi_{pot}(\delta^2 b^2_{i,j})\} \tag{5.2.14}$$

can be extended to an additive model with the following cost functional

$$C^{Charb}(u, b^2) = \{\sum_{(i,j)}[(\mathcal{K}u)_{i,j} - f^{\epsilon}_{i,j}]^2\} + \lambda\{\sum_{(i,j)} b^2_{i,j}[(D_x u)^2_{i,j} + (D_y u)^2_{i,j}] + \sum_{(i,j)} \psi_{pot}(\delta^2 b^2_{i,j})\}, \tag{5.2.15}$$

where $\lambda > 0$ is the pre-chosen regularisation parameter. This cost functional to-

gether with Algorithm 2.5.3 (alternating minimisation) defines a Charbonnier model as discussed in Chapter 2 [CBFAB97], with the name of "Additive counterpart of MMR" in this thesis. I will present numerical comparisons of these models in later sections.

## 5.3 A Model with Self-Adaptive Scaling Parameter

In this section, I introduce a new Model with Self-adaptive Scaling Parameter (MSSP), which includes a scaling parameter $\delta_n$ being self-adaptive in the algorithm. In addition to the weight function $b$ and the image intensity $u$, MSSP needs to update the scaling parameter $\delta$, and is equipped with a three-step alternating minimisation algorithm. This model is inspired by Abubakar *et al* [AvdBHB04], but can be put on a better theoretical footing, using the properties of the potential function $\varphi_{pot}$.

### 5.3.1 Model Formulation of MSSP

In MSSP, we construct a sequence $b_n^2$, $\delta_n^2$ and $u_n$ by applying a three step modified alternating minimisation algorithm to the following objective cost functional $C(u, b^2, \delta^2)$, which requires only semi-convergence in the minimisation of the image intensity $u$:

$$C^{MSSP}(u, b^2, \delta^2) = \{\sum_{(i,j)}[(\mathcal{K}u)_{i,j} - f^{\epsilon}_{i,j}]^2\} \cdot \{\sum_{(i,j)} b^2_{i,j}[(D_x u)^2_{i,j} + (D_y u)^2_{i,j}] + \sum_{(i,j)} \psi_{pot}(\delta^2 b^2_{i,j})\} \tag{5.3.1}$$

Note that the form of $C(u, b^2, \delta^2)$ in (5.3.1) remain the same as (5.2.14) in MMR. It may be verified from Table 2.3 that $C(u, b^2, \delta^2)$ is convex in $b^2$ and $\delta^2$. In order to extend the two-step modified alternating minimisation algorithm (*i.e.* Algorithm

5.2.1 for MMR) to a three step alternating minimisation algorithm and accommodate the self-adaptive scaling parameter $\delta_n$ in MSSP, we consider the following first order optimality conditions:

$$\frac{\partial C(u, b^2, \delta^2)}{\partial (b^2_{i,j})} = 0 \quad \Longleftrightarrow \quad \psi'_{pot}(\delta^2 b^2_{i,j})\delta^2 + (D_x u)^2_{i,j} + (D_y u)^2_{i,j} = 0, \ \forall \ i, j \quad (5.3.2)$$

$$\frac{\partial C(u, b^2, \delta^2)}{\partial (\delta^2)} = 0 \quad \Longleftrightarrow \quad \sum_{(i,j)} \psi'_{pot}(\delta^2 b^2_{i,j}) b^2_{i,j} = 0. \quad (5.3.3)$$

This motivates the following recurrence relations for updating the weight function $b^2$ and the scaling parameter $\delta^2$:

$$\psi'_{pot}(\delta^2_n (b^2_{n+1})_{i,j})\delta^2_n + (D_x u_n)^2_{i,j} + (D_y u_n)^2_{i,j} = 0, \ \forall \ i, j \quad (5.3.4)$$

to update $b^2_{n+1}$, and then

$$\sum_{(i,j)} \psi'_{pot}(\delta^2_{n+1} (b^2_{n+1})_{i,j})(b^2_{n+1})_{i,j} = 0 \quad (5.3.5)$$

for $\delta_{n+1}$. Note that the updating rule (5.3.4) is consistent with (5.2.1) in Algorithm 5.2.1 for MMR. This may be verified by the relationship between potential function $\psi$ and $\varphi$ via Remark 2.5.2 in Chapter 2. Explicit expressions for MSSP recurrence relations are presented in Table 5.1 for different forms of potential function $\psi_{pot}$.

We employ a three-step modified alternating minimisation algorithm for MSSP, which can be obtained naturally by adjusting the previous Algorithm 5.2.1 for MMR as follows:

**Algorithm 5.3.1.** ***Three-step modified alternating minimisation algorithm***
*Initialization:* $u_0^{M_0} = f^\epsilon$, $M_{inner}$ *and* $M_{outer}$ *chosen as in 5.2.1*
$\delta_0 =$ *arbitrarily large positive number, for the same reason as discussed previously in Section 3.2.3 for the original multiplicative model by Abubakar et al [AvdBHB04].*
*For* $0 \leq n \leq M_{outer}$, *Repeat*

***Outer loop:***

- *Fix $\delta = \delta_n$ and update $b_{n+1}$ according to (5.3.4)*
- *Fix $b = b_{n+1}$ and update $\delta_{n+1}$ according to (5.3.5)*

***Inner loop:*** *Fix $b = b_{n+1}$ and $\delta = \delta_{n+1}$ and iterate for $u$ on the superscript $m$*
*Initialization: $u_{n+1}^0 = u_n^{M_n}$.*
*If $1 \leq m \leq M_{inner}$, then*

- *Update image intensity $u_{n+1}^{m+1}$ as in Algorithm 5.2.1, but replacing the regularisation term in the old formulation by the new one in the objective functional (5.3.1);*

*The termination criteria for both the inner and outer loops are the same as in Algorithm 5.2.1.*

| Potential Function | Recurrence Relations |
|---|---|
| HL | $(b_{n+1}^2)_{i,j} = \frac{1}{(D_x u_n)_{i,j}^2 + (D_y u_n)_{i,j}^2 + \delta_n^2} \ \forall\ i,j = 1,2,...N$<br>$\delta_{n+1}^2 = \frac{N^2}{\sum_{(i,j)} (b_{n+1}^2)_{i,j}}$ |
| GM | $(b_{n+1}^2)_{i,j} = \frac{\delta_n^2}{[(D_x u_n)_{i,j}^2 + (D_y u_n)_{i,j}^2 + \delta_n^2]^2} \ \forall\ i,j = 1,2,...N.$<br>$\delta_{n+1} = \frac{\sum_{(i,j)} \sqrt{(b_{n+1}^2)_{i,j}}}{\sum_{(i,j)} (b_{n+1}^2)_{i,j}}$ |
| HS | $(b_{n+1}^4)_{i,j} = \frac{1}{\delta_n^2 [(D_x u_n)_{i,j}^2 + (D_y u_n)_{i,j}^2 + \delta_n^2]} \ \forall\ i,j = 1,2,...N.$<br>$\delta_{n+1}^4 = \frac{\sum_{(i,j)} \frac{1}{(b_{n+1}^2)_{i,j}}}{\sum_{(i,j)} (b_{n+1}^2)_{i,j}}$ |

Table 5.1: Recurrence Relations for different types of potential functions

### 5.3.2 Additive Counterpart of MSSP

Similarly to MMR, there is an additive model corresponding to MSSP, with the following objective cost functional

$$C^{Add}(u,b,\delta) = \{\sum_{(i,j)} [(\mathcal{K}u)_{i,j} - f_{i,j}^{\epsilon}]^2\} + \lambda\{\sum_{(i,j)} b_{i,j}^2 [(D_x u)_{i,j}^2 + (D_y u)_{i,j}^2] + \sum_{(i,j)} \psi_{pot}(\delta^2 b_{i,j}^2)\} \tag{5.3.6}$$

Here $\lambda > 0$ is the pre-chosen regularisation parameter. We define the following three step alternating minimisation algorithm for the additive counterpart of MSSP to complete this model:

**Algorithm 5.3.2.** ***Three step alternating minimisation for the additive counterpart of MSSP***
*Step 0: Initialization: choose* $u^0 = f^\epsilon$, $M_{outer}$, *tol and* $\delta_0^2$ *to start with;*
*Repeat*
*Step 1: Update* $b_{n+1}^2$ *according to (5.3.4);*
*Step 2: Update* $\delta_{n+1}^2$ *according to (5.3.5);*
*Step 3: Update* $u_{n+1}$ *by taking the unique minimiser of* $C^{Add}(u, b_{n+1}^2, \delta_{n+1}^2)$*;*
*Termination criterion: If*

$$||u_{n+1} - u_n||_2^2 < tol$$

*or* $n > M_{outer}$*, then stop.*

### 5.3.3 Semi-convergence of MSSP and its Additive Counterpart

In this section we take a closer look at the optimality conditions (5.3.2) and (5.3.3), which work for both multiplicative cost functional and its additive counterpart $C^{Add}(u, b^2, \delta^2)$. Consider the minimisation of the objective cost functional for a given image intensity $u$. For $u \neq constant$, there exists $(i_0, j_0) \in \{(i,j)\big|1 \leq i, j \leq N\}$ such that $(D_x u)_{i_0,j_0}^2 + (D_y u)_{i_0,j_0}^2 \neq 0$. Hence we can conclude $\delta^2 \neq 0$, and from the coupled system (5.3.2) and (5.3.3) we can obtain

$$0 = \sum_{(i,j)} \psi'_{pot}(\delta^2 b_{i,j}^2) b_{i,j}^2 = -\frac{1}{\delta^2} \sum_{(i,j)} [(D_x u)_{i,j}^2 + (D_y u)_{i,j}^2] b_{i,j}^2 \leq 0, \tag{5.3.7}$$

where equality takes place at $b_{i,j}^2 = 0, \forall (i,j) \in \{(i,j)\big|(D_x u)_{i,j}^2 + (D_y u)_{i,j}^2 \neq 0\}$. This

means that for each given image intensity $u$, the $\hat{b}^2$ and $\hat{\delta}^2$ such that

$$\tilde{C}(u, \hat{b}^2, \hat{\delta}^2) = \min_{b^2, \delta^2} C(u, b^2, \delta^2) \tag{5.3.8}$$

are not necessarily well-defined. To see this, note that

$$\sum_{(i,j)} (\hat{b}_u^2)_{i,j} [(D_x u)_{i,j}^2 + (D_y u)_{i,j}^2] = 0$$

for both of the multiplicative and the additive model. Due to the non-negativity of the objective cost functionals $C^{MSSP}(u, b^2, \delta^2)$ and $C^{Add}(u, b^2, \delta^2)$ in (5.3.1) and (5.3.6), we are now left with the minimisation of term $\sum_{(i,j)} \psi_{pot}(\delta^2 b_{i,j}^2)$. Recall that $\psi_{pot}(\omega) \geq 0$ with $\psi_{pot}(1) = 0$ as can be checked from Table 2.3. Therefore, the minimisation can potentially direct $\hat{\delta}^2$ to arbitrarily large values ,to accommodate $\delta^2 b_{i,j}^2 = 1$ for those combination of $(i, j)$ such that $b_{i,j}^2 = 0$.

However, this is neither desirable, nor where MSSP directs its solution to. In both algorithms for MSSP and its additive counterpart, we do not solve for the coupled system, but instead minimise the objective functional in terms of $b_{i,j}^2$ and $\delta^2$ alternately, keeping the other variable constant. In other words, only one equation of the system of recurrence relations (5.3.2) and (5.3.3) is satisfied at each time. This means that for both MSSP and its additive counterpart, Algorithm 5.3.2 does not direct their solutions to the global minimum. Instead, they both exhibit semi-convergence behaviour similar to other multiplicative models.

## 5.4 Convergence Analysis for MMR

In this section, we show the convergence of the outer loop with index $n$ in MMR, towards completing proof of convergence for this method. We begin from simple results on the convergence of the objective cost functional, and later discuss the convergence properties of the weight function $b_n^2$ and image intensity $u_n$ under reasonable assumptions.

### 5.4.1 Convergence of Cost Functional $C(u_n^{M_n}, b_n^2)$

Previously we have shown the monotonical decrease of the objective cost functional $\{C(u_{n+1}^m, b_{n+1}^2)\}_{m\in\mathbb{N}}$ in the inner loop for each fixed $n \in \mathbb{N}$ in (5.2.13):

$$C(u_n^{m-1}, b_n^2) \geq C(u_n^m, b_n^2),\ \forall 1 \leq m \leq M_n,$$

where $M_n$ denotes the total number of iterations conducted at the $n$th step of the outer loop. As

$$u_{n+1}^0 = u_n^{M_n},\ \forall n \in \mathbb{N}$$

it follows that

$$C(u_n^0, b_n^2) \geq C(u_n^{M_n}, b_n^2) = C(u_{n+1}^0, b_n^2),\quad \forall n \in \mathbb{N} \tag{5.4.1}$$

Note that the weight function $b_{n+1}^2$ is updated via a closed form solution of

$$b_{n+1}^2 = \arg\min_{b^2}[C(u_n^{M_n}, b^2)],$$

which leads to

$$C(u_n^{M_n}, b_n^2) \geq C(u_n^{M_n}, b_{n+1}^2) = C(u_{n+1}^0, b_{n+1}^2),\ \forall n \in \mathbb{N} \tag{5.4.2}$$

Hence by (5.4.1) and (5.4.2) the objective cost functional $\{C(u_n^0, b_n^2)\}_{n\in\mathbb{N}}$ decreases monotonically in the outer loop, *i.e.*

$$C(u_n^0, b_n^2) \geq C(u_{n+1}^0, b_{n+1}^2),\ \forall n \in \mathbb{N}. \tag{5.4.3}$$

It is obvious that $C(u_n^0, b_n^2)$ stays positive for all $n \in \mathbb{N}$. As a monotonically decreasing sequence bounded below by 0, then the sequence of the objective cost functional

$\{C(u_n^0, b_n^2)\}_{n\in\mathbb{N}}$ certainly converges.
From (5.1.3), we have

$$C(u_n^0, b_n^2) = C(u_{n-1}^{M_{n-1}}, b_n^2) = \min_{b^2} C(u_{n-1}^{M_{n-1}}, b^2) = \bar{C}(u_{n-1}^{M_{n-1}}) = \bar{C}(u_n^0), \qquad (5.4.4)$$

and the convergence of sequence $\{\bar{C}(u_n^0)\}_{n\in\mathbb{N}}$ follows from the convergence of $\{C(u_n^0, b_n^2)\}_{n\in\mathbb{N}}$. In fact, we have the following descending chain which joins both inner and outer loops:

$$C(u_n^{M_n}, b_n^2) \geq C(u_n^{M_n}, b_{n+1}^2) = C(u_{n+1}^0, b_{n+1}^2) \geq C(u_{n+1}^{M_{n+1}}, b_{n+1}^2) \geq ... \qquad (5.4.5)$$

This result will be used in later sections for further convergence analysis.

### 5.4.2 Asymptotic Properties of Weight Function $b_n^2$

Inspired by the method used by Charbonnier *et al* [CBFAB97] for an an additive type of regularisation, we extend their proof to MMR, and show that $|b_{n+1}^2 - b_n^2| \to 0$, as $n \to \infty$ with the following assumption:

**Assumption 5.4.1.** *If $u_n^{M_n} \nrightarrow u^\dagger = (\mathcal{K}^*\mathcal{K})^{-1}\mathcal{K}^* f^\epsilon$, then there exists $\kappa \in \mathbb{R}^+$ such that*

$$\sum_{(i,j)} [(\mathcal{K}u_n^{M_n})_{i,j} - f_{i,j}^\epsilon]^2 \geq \kappa, \ as \ n \to \infty. \qquad (5.4.6)$$

This assumption essentially states that MMR exhibits semi-convergence and behaves like an iterative method. The final solution is bounded away from the Moore-Penrose generalised inverse. In this section, we aim to show

$$|(b_{n+1}^2)_{i,j} - (b_n^2)_{i,j}| \to 0, \ \forall 1 \leq i, j \leq N. \qquad (5.4.7)$$

Given the form (5.1.2) of the objective cost functional in MMR, consider

$$C(u_n^{M_n}, b_n^2) - C(u_n^{M_n}, b_{n+1}^2) = \{\sum_{(i,j)}[(\mathcal{K}u_n^{M_n})_{i,j} - f_{i,j}^{\epsilon}]^2\} \cdot \sum_{(i,j)}\{p_{i,j}[(b_n^2)_{i,j}] - p_{i,j}[(b_{n+1}^2)_{i,j}]\},$$

where

$$p_{i,j}(\omega) := \omega[(D_x u)_{i,j}^2 + (D_y u)_{i,j}^2] + \psi_{pot}(\delta^2 \omega). \tag{5.4.8}$$

Apply Taylor expansion for $p_{i,j}(\omega)$ at $\omega = (b_{n+1}^2)_{i,j}$:

$$C(u_n^{M_n}, b_n^2) - C(u_n^{M_n}, b_{n+1}^2) = \{\sum_{(i,j)}[(\mathcal{K}u_n^{M_n})_{i,j} - f_{i,j}^{\epsilon}]^2\} \cdot$$
$$\{\sum_{(i,j)} p'_{i,j}[(b_{n+1}^2)_{i,j}][(b_n^2)_{i,j} - (b_{n+1}^2)_{i,j}] + \frac{1}{2}\sum_{(i,j)} p''_{i,j}[(c_{n+1})_{i,j}][(b_n^2)_{i,j} - (b_{n+1}^2)_{i,j}]^2\},$$

with $(c_{n+1})_{i,j} \in \big((b_n^2)_{i,j}, (b_{n+1}^2)_{i,j}\big)$. Since $b_{n+1}^2 = \arg\min_{b^2}[C(u_n^{M_n}, b^2)]$, we have

$$\frac{\partial}{\partial(b_{i,j}^2)} C(u_n^{M_n}, b_{i,j}^2)\big|_{b^2 = b_{n+1}^2} = \{\sum_{(i,j)}[(\mathcal{K}u_n^{M_n})_{i,j} - f_{i,j}^{\epsilon}]^2\} \cdot p'_{i,j}[(b_{n+1}^2)_{i,j}] = 0, \ \forall (i,j).$$

This means that the first order terms in the Taylor expansion vanishes, *i.e.*

$$\{\sum_{(i,j)}[(\mathcal{K}u_n^{M_n})_{i,j} - f_{i,j}^{\epsilon}]^2\} \cdot \{\sum_{(i,j)} p'_{i,j}[(b_{n+1}^2)_{i,j}][(b_n^2)_{i,j} - (b_{n+1}^2)_{i,j}]\} = 0, \ \forall 1 \le i,j \le N.$$

As a result,

$$\begin{aligned} & C(u_n^{M_n}, b_n^2) - C(u_n^{M_n}, b_{n+1}^2) \\ = \ & \{\textstyle\sum_{(i,j)}[(\mathcal{K}u_n^{M_n})_{i,j} - f_{i,j}^{\epsilon}]^2\} \cdot \{\frac{1}{2}\sum_{(i,j)} p''_{i,j}[(c_{n+1})_{i,j}][(b_n^2)_{i,j} - (b_{n+1}^2)_{i,j}]^2\} \end{aligned} \tag{5.4.9}$$

Recall from previous section that $C(u_n, b_n^2)$ is a descending chain as described in (5.4.5), the left hand side of (5.4.9) is always positive, and eventually tends to 0 as $n \to \infty$. Together with eq (5.4.6), in order to force $[(b_n^2)_{i,j} - (b_{n+1}^2)_{i,j}]^2 \to 0$, one just

needs to find a lower bound for $p''_{i,j}[(c_{n+1})_{i,j}]$ for all possible choices of $(c_{n+1})_{i,j}$. In other words, to show that $\exists\Lambda > 0$ such that

$$p''_{i,j}(\omega) \geq \Lambda,\ \forall 1 \leq i,j \leq N^2,\ \forall\omega > 0 \tag{5.4.10}$$

We can deduce from (5.4.8)

$$p''_{i,j}(\omega) = \delta^4 \psi''_{pot}(\delta^2\omega). \tag{5.4.11}$$

With change of variable

$$\omega^* = \delta^2\omega \tag{5.4.12}$$

(5.4.11) becomes

$$\psi''_{pot}(\omega^*) = p''_{i,j}(\omega^*),\quad \forall\omega > 0.$$

Note that this expression corresponds in our notation to (C.0.10) in Appendix C. From this point, it is possible to continue with the proof by Charbonnier *et al* [CBFAB97] in Appendix C from (C.0.10) onwards, and eventually show

$$\left|\psi''_{pot}(\omega^*)\right| \geq \frac{1}{C},$$

for some $C > 0$ as shown in (C.0.18) of Appendix C. This leads to $\Lambda = \frac{1}{C}$, and

$$C(u_n^{M_n}, b_n^2) - C(u_n^{M_n}, b_{n+1}^2) \geq \frac{1}{2}\kappa\Lambda \sum_{(i,j)} [(b_n^2)_{i,j} - (b_{n+1}^2)_{i,j}]^2.$$

$C(u_n^{M_n}, b_n^2) - C(u_n^{M_n}, b_{n+1}^2) \to 0$ due to convergence of $C(u_n^{M_n}, b_n^2)$. Hence $[(b_n^2)_{i,j} - (b_{n+1}^2)_{i,j}] \to 0$.

## 5.5 Convergence Analysis for MSSP and its Additive Counterpart

### 5.5.1 Convergence of Cost Functional $C(u_n^{M_n}, b_n^2, \delta_n^2)$

Firstly we show monotonic decrease of objective cost functional $C^{MSSP}(u_n^{M_n}, b_n^2, \delta_n^2)$ and $C^{Add}(u_n, b_n^2, \delta_n^2)$ for both MSSP and its additive counterpart:
In MSSP, three step modified alternating minimisation algorithm gives

$$\begin{aligned} b_{n+1}^2 &= \min_{b^2} C^{MSSP}(u_n^{M_n}, b^2, \delta_n^2) \implies C^{MSSP}(u_n^{M_n}, b_n^2, \delta_n^2) \geq C^{MSSP}(u_n^{M_n}, b_{n+1}^2, \delta_n^2) \\ \delta_{n+1}^2 &= \min_{\delta^2} C^{MSSP}(u_n^{M_n}, b_{n+1}^2, \delta^2) \implies C^{MSSP}(u_n^{M_n}, b_{n+1}^2, \delta_{n+1}^2) \geq C^{MSSP}(u_n^{M_n}, b_{n+1}^2, \delta_n^2) \end{aligned}$$

Further more,

$$C^{MSSP}(u_n^{M_n}, b_{n+1}^2, \delta_{n+1}^2) = C^{MSSP}(u_{n+1}^0, b_{n+1}^2, \delta_{n+1}^2) \geq C^{MSSP}(u_{n+1}^{M_{n+1}}, b_{n+1}^2, \delta_{n+1}^2)$$

is implied by Proposition 5.2.2 as discussed in Section 5.2. Hence we have

$$C^{MSSP}(u_n^{M_n}, b^2, \delta_n^2) \geq C^{MSSP}(u_{n+1}^{M_{n+1}}, b_{n+1}^2, \delta_{n+1}^2)$$

For its additive counterpart, similarly one can obtain

$$\begin{aligned} b_{n+1}^2 &= \min_{b^2} C^{Add}(u_n, b^2, \delta_n^2) \implies C^{Add}(u_n, b_n^2, \delta_n^2) \geq C^{Add}(u_n, b_{n+1}^2, \delta_n^2) \\ \delta_{n+1}^2 &= \min_{\delta^2} C^{Add}(u_n, b_{n+1}^2, \delta^2) \implies C^{Add}(u_n, b_{n+1}^2, \delta_{n+1}^2) \geq C^{Add}(u_n, b_{n+1}^2, \delta_n^2) \\ u_{n+1} &= \min_{u} C^{Add}(u, b_{n+1}^2, \delta_{n+1}^2) \implies C^{Add}(u_{n+1}, b_{n+1}^2, \delta_{n+1}^2) \end{aligned}$$

Hence

$$C^{Add}(u_n, b^2, \delta_n^2) \geq C^{Add}(u_{n+1}, b_{n+1}^2, \delta_{n+1}^2)$$

By construction

$$C^{MSSP}(u_n^{M_n}, b_n^2, \delta_n^2) \geq 0,$$
$$C^{Add}(u_n, b_n^2, \delta_n^2) \geq 0,$$

Hence $\{C^{MSSP}(u_n^{M_n}, b_n^2, \delta_n^2)\}_{n\in\mathbb{N}}$ and $\{C^{Add}(u_n, b_n^2, \delta_n^2)\}_{n\in\mathbb{N}}$ both converge (monotonically decreasing and bounded below).

### 5.5.2 Asymptotic Properties of Complementary Variables $b_n$ and $\delta_n$

**Case of HL type potential function**

In this section I will discuss convergence properties of the complementary variables $b_n$ and $\delta_n$ for both MSSP and its additive counterpart. Similarly to the previous MMR model as discussed in Section 5.4.2, Assumption 5.4.1 will be assumed for the multiplicative model MSSP, and we consider first the case when the potential functions $\varphi_{pot}$ and $\psi_{pot}$ take the HL form by Hebert and Leahy [LH89]. The argument may be extended from Charbonnier's proofs [CBFAB97] as presented in Chapter 2. Consider Taylor expansion in variable $b^2$ at $b^2 = b_{n+1}^2$. We have

$$\begin{aligned} & C^{MSSP}(u_n, b_n^2, \delta_n^2) - C^{MSSP}(u_n, b_{n+1}^2, \delta_n^2) \\ = \; & \frac{\delta_n^4}{2}\{\sum_{(i,j)}[(\mathcal{K}u)_{i,j} - f_{i,j}^{\epsilon}]^2\} \cdot \sum_{(i,j)} \psi''_{pot}(\delta_n^2 (c_1^2)_{i,j})[(b_n^2)_{i,j} - (b_{n+1}^2)_{i,j}]^2, \quad (5.5.1) \end{aligned}$$

for MSSP, and

$$C^{Add}(u_n, b_n^2, \delta_n^2) - C^{Add}(u_n, b_{n+1}^2, \delta_n^2) = \frac{\lambda\delta_n^4}{2} \sum_{(i,j)} \psi''_{pot}(\delta_n^2 (c_2^2)_{i,j})[(b_n^2)_{i,j} - (b_{n+1}^2)_{i,j}]^2, \quad (5.5.2)$$

for its additive counterpart. In (5.5.2), $\lambda > 0$ is the regularisation parameter; in both (5.5.1) and (5.5.2), $(c_1^2)_{i,j}, (c_2^2)_{i,j} \in [(b_{n+1}^2)_{i,j}, (b_n^2)_{i,j}]$. The first order term vanishes

due to the recurrence relation (5.3.4) which results from the first order optimality condition with stationary point at $b^2_{i,j} = (b^2_{n+1})_{i,j}$, and the extra $\delta^4_n$ term comes from the chain rule. Since the potential function has the form

$$\psi_{HL}(\omega) = \omega - \log(\omega) - 1, \quad \psi''_{HL}(\omega) = \frac{1}{\omega^2},$$

we can calculate that

$$\delta^4 \psi''_{HL}(\delta^2\omega^2) = \frac{1}{\omega} = \psi''_{HL}(\omega)$$

with no dependence on parameter $\delta^2$. This allows Taylor expansions (5.5.1) and (5.5.3) in MSSP to be expressed as

$$\begin{aligned}
& C^{MSSP}(u_n, b^2_n, \delta^2_n) - C^{MSSP}(u_n, b^2_{n+1}, \delta^2_n) \\
= \; & \frac{1}{2}\{\sum_{(i,j)}[(\mathcal{K}u)_{i,j} - f^{\epsilon}_{i,j}]^2\}\psi''_{pot}((c^2_1)_{i,j})[(b^2_n)_{i,j} - (b^2_{n+1})_{i,j}]^2, \\
& C^{MSSP}(u_n, b^2_{n+1}, \delta^2_n) - C^{MSSP}(u_n, b^2_{n+1}, \delta^2_n) \\
= \; & \frac{1}{2}\{\sum_{(i,j)}[(\mathcal{K}u)_{i,j} - f^{\epsilon}_{i,j}]^2\}\psi''_{pot}(d^2_1)(\delta^2_n - \delta^2_{n+1})^2,
\end{aligned}$$

In the additive counterpart of MSSP, (5.5.2) and (5.5.4) become

$$\begin{aligned}
C^{Add}(u_n, b^2_n, \delta^2_n) - C^{Add}(u_n, b^2_{n+1}, \delta^2_n) &= \frac{\lambda}{2}\sum_{(i,j)}\psi''_{pot}((c^2_2)_{i,j})[(b^2_n)_{i,j} - (b^2_{n+1})_{i,j}]^2, \\
C^{Add}(u_n, b^2_{n+1}, \delta^2_n) - C^{Add}(u_n, b^2_{n+1}, \delta^2_{n+1}) &= \frac{\lambda N^2}{2}\psi''_{pot}(d^2_2)(\delta^2_n - \delta^2_{n+1})^2,
\end{aligned}$$

where $(c^2_1)_{i,j}, (c^2_2)_{i,j} \in [(b^2_{n+1})_{i,j}, (b^2_n)_{i,j}]$, and $d^2_1, d^2_2 \in [\delta^2_{n+1}, \delta^2_n]$. Note that the complementary variables $b^2$ and $\delta^2$ are separated in both MSSP and its additive counterpart. As a consequence, the potential functions in these cost functionals will admit lower bounds $\psi''_{pot}((c^2_1)_{i,j}) \geq D_1,\ \psi''_{pot}((c^2_2)_{i,j}) \geq D_2,\ \forall i,j = 1,2,...N$ and $\psi''_{pot}(d^2_1) \geq D_3,\ \psi''_{pot}(d^2_2) \geq D_4$ for some $D_1,\ D_2,\ D_3,\ D_4 > 0$, and force $[(b^2_n)_{i,j} - (b^2_{n+1})_{i,j}]^2 \to 0$

as well as $(\delta_n^2 - \delta_{n+1}^2)^2 \to 0$ as $n \to \infty$ in both MSSP and its additive counterpart (with Assumption 5.4.1).

**Case of different potential functions**

Unfortunately, the extension of Charbonnier's proof [CBFAB97] and MMR model fails in the case of other potential functions beyond this point: despite the fact that it still holds that both left hand side of (5.5.1) and (5.5.2) tend to 0 as $n \to \infty$, it is no longer possible to force $[\hat{b}_{i,j}^2 - (b_{n+1}^2)_{i,j}]^2 \to 0$ on the right hand of these two expressions in MSSP. The argument fails due to the presence of the extra term $\delta_n^4$ which causes failure in the change of variable step in (5.4.12): as the scaling parameter $\delta_n^2$ is self-adaptive in MSSP, its value cannot be classified easily for $n \to \infty$, as there is little information about it at this stage. In fact, $\delta_n^2 \to 0$ makes perfect sense in image reconstruction. As we can see from Table 5.1, the scaling parameter $\delta_n^2$ together with the finite difference $[(D_x u)_{i,j}^2 + (D_y u)_{i,j}^2]$ determines the weight function $b_n^2$ ,hence controls the spatial regularisation effect at each pixel $(i, j)$. As $\delta_n^2 \to 0$, the regularisation effect is predominantly determined by the finite difference term: large regularisation is applied to pixels with small finite difference ("interior points"), while small regularisation is applied to pixels with large finite difference ("edge points").

Similarly, consider Taylor expansion in variable $\delta^2$ at $\delta^2 = \delta_{n+1}^2$ for both MSSP and its additive counterpart

$$\begin{aligned} & C^{MSSP}(u_n, b_{n+1}^2, \delta_n^2) - C^{MSSP}(u_n, b_{n+1}^2, \delta_n^2) \\ = \; & \frac{1}{2}\{\sum_{(i,j)}[(\mathcal{K}u)_{i,j} - f_{i,j}^{\epsilon}]^2\}\{\sum_{(i,j)} \psi''_{pot}[d_1^2 (b_{n+1})_{i,j}^2](b_{n+1})_{i,j}^4 (\delta_n^2 - \delta_{n+1}^2)^2\} \; , \end{aligned} \tag{5.5.3}$$

and

$$C^{Add}(u_n, b_{n+1}^2, \delta_n^2) - C^{Add}(u_n, b_{n+1}^2, \delta_{n+1}^2) = \frac{\lambda}{2} \sum_{(i,j)} \psi''_{pot}[d_2^2 (b_{n+1}^2)_{i,j}](b_{n+1}^4)_{i,j} (\delta_n^2 - \delta_{n+1}^2)^2, \tag{5.5.4}$$

where $d_1^2, d_2^2 \in [\delta_{n+1}^2, \delta_n^2]$. Due to the presence of self-adaptive term $(b_{n+1})_{i,j}^4$ whose value cannot be classified as $n \to \infty$ at this stage, no further conclusion may be drawn regarding the individual convergence property of complementary variable $\delta_n^2$ or $b_n^2$.

Although it is not possible to extend the proof by Charbonnier *et al* [CBFAB97] to show convergence of complementary variable $\delta_n^2$ or $b_n^2$ individually, it is however still possible to draw conclusions for product $\delta_n^2 b_n^2$ with the uniform boundedness of

$$\begin{aligned}
&\psi''_{pot}(\delta_n^2(c_1^2)_{i,j}) \geq \frac{1}{C_{i,j}^1} \quad , \quad \psi''_{pot}[d_1^2(b_{n+1}^2)_{i,j}] \geq \frac{1}{C'}, \\
&\psi''_{pot}(\delta_n^2(c_2^2)_{i,j}) \geq \frac{1}{C_{i,j}^2} \quad , \quad \psi''_{pot}[d_2^2(b_{n+1}^2)_{i,j}] \geq \frac{1}{C''}, \ \forall\ i,j = 1,2,...N,
\end{aligned}$$

for some $C_{i,j}^1, C_{i,j}^2, C', C'' > 0$ as shown in Appendix C (C.0.10) onwards. Let us re-assess (5.5.1) and (5.5.3) in MSSP model with Assumption 5.4.1, and (5.5.2) and (5.5.4) in its additive counterpart. The left hand side of all these four equations tends to 0 as $n \to \infty$. Hence it follows from these four equations that

$$\begin{aligned}
\sum_{(i,j)} [\delta_n^2 (b_n^2)_{i,j} - \delta_n^2 (b_{n+1}^2)_{i,j}]^2 \to 0, \quad n \to \infty \\
\sum_{(i,j)} [\delta_n^2 (b_{n+1}^2)_{i,j} - \delta_{n+1}^2 (b_{n+1}^2)_{i,j}]^2 \to 0, \quad n \to \infty
\end{aligned}$$

By triangle inequality, these two equations lead to

$$\begin{aligned}
&\textstyle\sum_{(i,j)} [\delta_{n+1}^2 (b_{n+1}^2)_{i,j} - \delta_n^2 (b_n^2)_{i,j}]^2 \\
\leq\ &\textstyle\sum_{(i,j)} [\delta_n^2 (b_n^2)_{i,j} - \delta_n^2 (b_{n+1}^2)_{i,j}]^2 + \sum_{(i,j)} [\delta_n^2 (b_{n+1}^2)_{i,j} - \delta_{n+1}^2 (b_{n+1}^2)_{i,j}]^2 \quad \to 0, \quad n \to \infty
\end{aligned}$$

Hence $[\delta_{n+1}^2 (b_{n+1}^2)_{i,j} - \delta_n^2 (b_n^2)_{i,j}]^2 \to 0$ as $n \to \infty$.

## 5.6 Numerical Implementation for MMR and MSSP

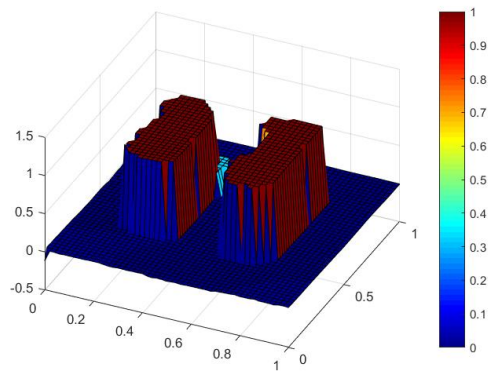
### 5.6.1 HL Potential Function

In this section I present numerical results for MMR and MSSP model both with HL type of potential function. To demonstrate their effectiveness, I firstly test them on the two-block image (1.3.2) for different noise amplitude and nature, and then compare them with Abubakar's multiplicative model [AvdBHB04]. Quantitative and qualitative results are presented in Table 5.2, Fig 5.2 and Fig 5.3 below, with scaling parameter in MMR set to be $\delta = 1$.

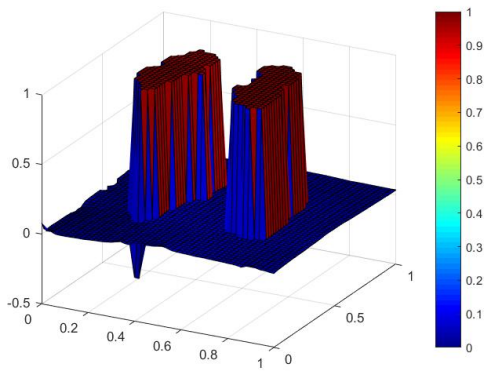
| Noise type | Model by | SSIM | PSNR |
|---|---|---|---|
| Gaussian | Abubakar | 0.68 | 19.38 |
| Gaussian | MMR | 0.63 | 17.45 |
| Gaussian | MSSP | 0.65 | 17.60 |
| K ($\alpha = 10$) (asymm bell) | Abubakar | 0.71 | 19.54 |
| K ($\alpha = 10$) (asymm bell) | MMR | 0.63 | 16.62 |
| K ($\alpha = 10$) (asymm bell) | MSSP | 0.61 | 17.02 |
| K ($\alpha = 0.5$)(spiky) | Abubakar | 0.70 | 19.64 |
| K ($\alpha = 0.5$)(spiky) | MMR | 0.64 | 16.68 |
| K ($\alpha = 0.5$)(spiky) | MSSP | 0.60 | 17.18 |

Table 5.2: SSIM and PSNR of the Reconstruction by Abubakar, MMR and the MSSP model (All noise implemented are at 10% level)

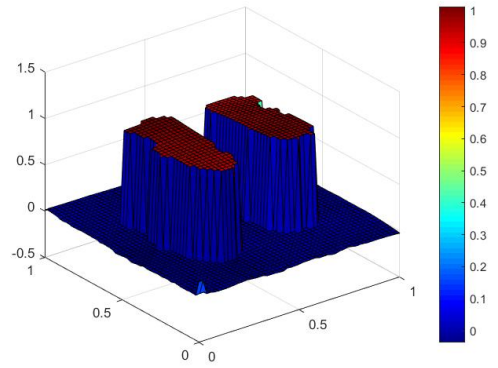
Looking at the results in Fig 5.2 and Fig 5.3 and those in Fig 3.4 and Fig 3.6 presented previously, we observe qualitatively from Fig 5.2 and Fig 5.3 that both MMR and MSSP model inherit the advantages of Abubakar's model [AvdBHB04] in terms of (a) robustness against noise with different nature; (b) leading to flat contours in the reconstructed image which advantages piecewise constant images particularly; and (c) capability of reconstructing the main image feature from high level of noise. However, both models exhibit quantitative behaviour weaker than Abubakar's multiplicative model in terms of metric SSIM and PSNR index, as illustrated in Table 5.2. For MMR, the explanation lies behind the tuning of parameter $\delta^2$. In other words, the self-adjustment of parameter $\delta$ contributes to the model in a non-negligible way,



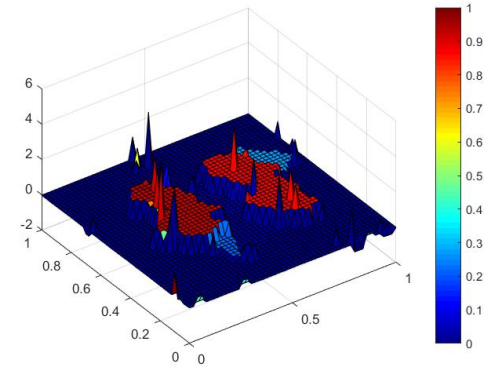
(a) Gaussian 10%



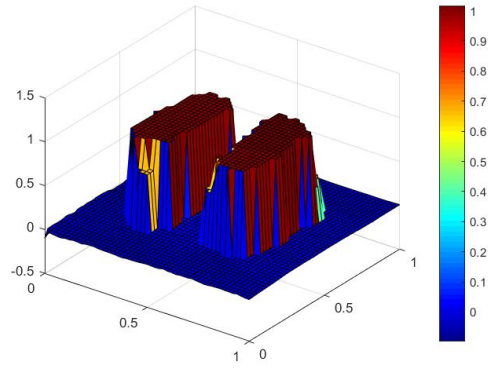
(b) Gaussian 50%



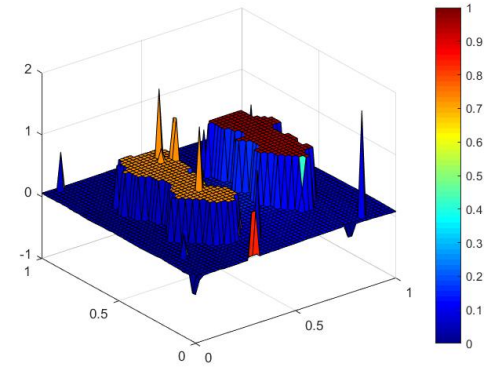
(c) K ($\alpha = 0.5$) (spiky) 10%



(d) K ($\alpha = 0.5$) (spiky) 50%

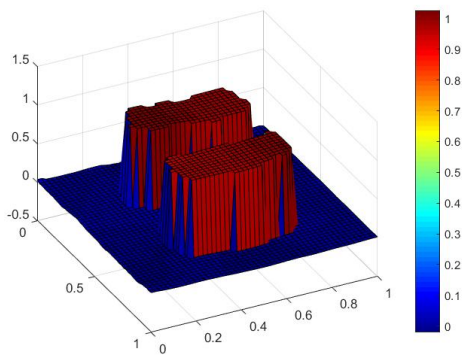


(e) K ($\alpha = 10$)(asymm bell) 10%

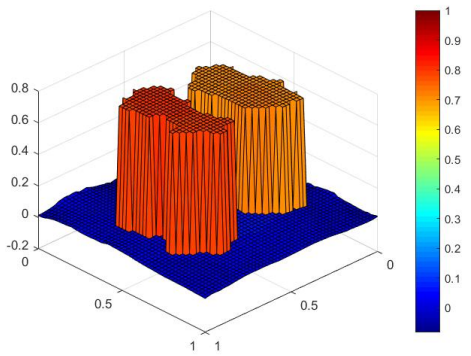


(f) K ($\alpha = 10$)(asymm bell) 50%

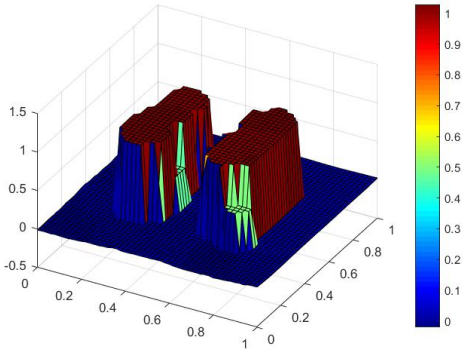
Figure 5.2: Reconstructed images by MMR for various noise distributions at amplitude 10% and 50%



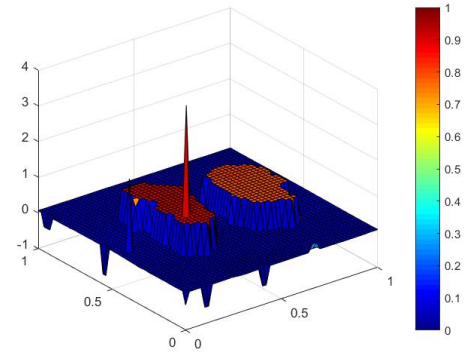
(a) Gaussian 10%



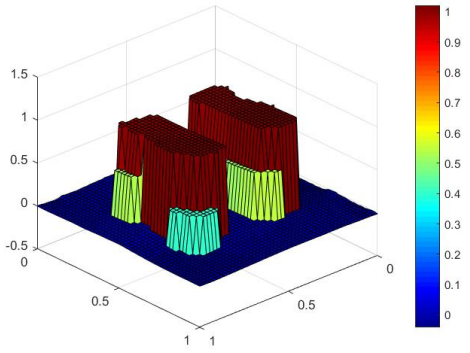
(b) Gaussian 50%



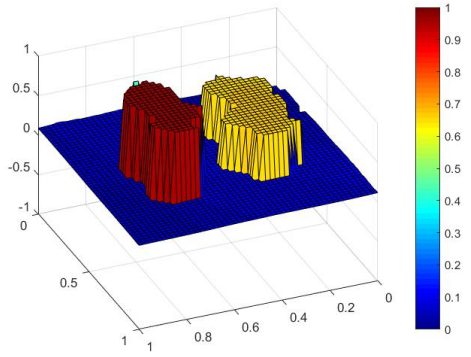
(c) K ($\alpha = 0.5$) (spiky) 10%



(d) K ($\alpha = 0.5$) (spiky) 50%



(e) K ($\alpha = 10$)(asymm bell) 10%



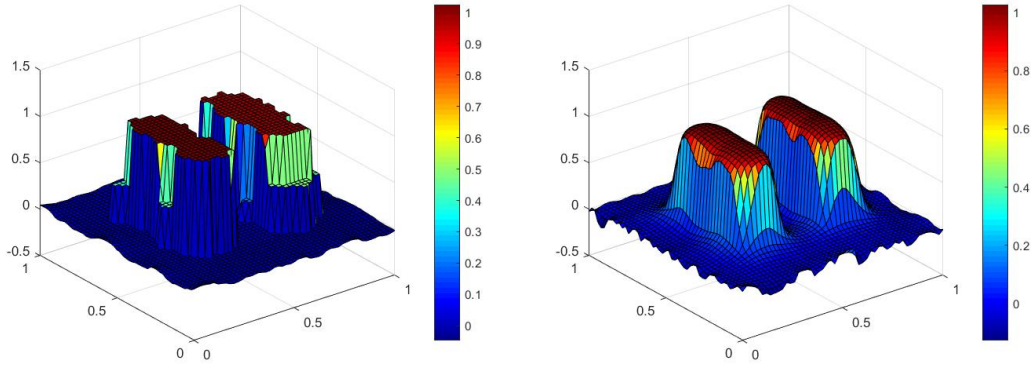
(f) K ($\alpha = 10$)(asymm bell) 50%

Figure 5.3: Reconstructed images by the MSSP model with different noise distributions at 10% and 50% level)

and keeping it constant jeopardizes the model. This motivates the formulation of MSSP, a second multiplicative model that admits a self-adaptive scaling parameter $\delta^2$ in the algorithm. Unfortunately, MSSP fails to achieve this goal due to the way that parameter $\delta_n$ is inserted. This calls for a different objective cost functional and a new set of recurrence relations for further numerical improvement. In the next chapter, I will provide a third formulation which continues with the three step minimization setting (EMM model), as well as leading to reconstructions exceeding the performance of Abubakar's model in [AvdBHB04].

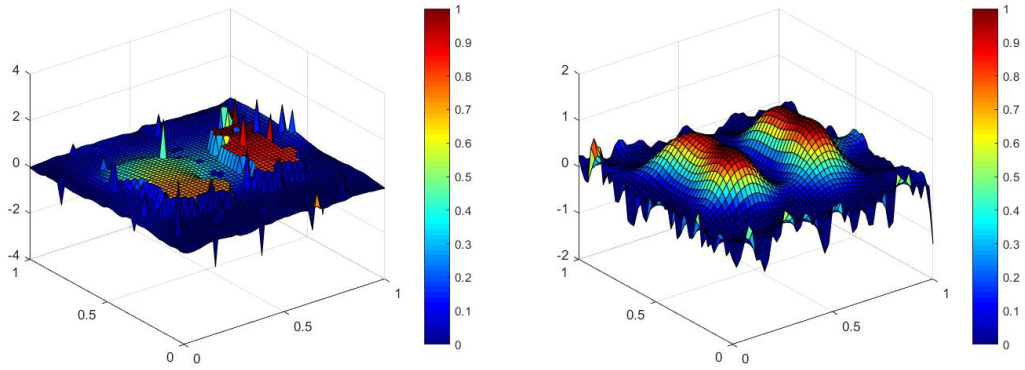
### 5.6.2 Other Types of Potential Functions

This section explores performance of MMR and MSSP model with potential function in the form of $\varphi_{GM}$ and $\varphi_{HS}$. With the same two-block test image (1.3.2), it may be observed from Fig 5.4-Fig 5.7 that (a) for both MMR and MSSP, $\varphi_{GM}$ leads to sharper edges of the reconstruction, while $\varphi_{HS}$ tends to smooth edges further; (b) both of these two potential functions have weaker performance in the case of highly noisy data compared with $\varphi_{HL}$ by Hebert and Leahy [LH89]: $\varphi_{GM}$ leads to more spurious spikes in the reconstruction, while $\varphi_{GM}$ smears out the main blocks to a large extent; (c) MSSP appears to be less robust than MMR at highly noisy date as shown in Fig 5.7: $\varphi_{GM}$ leads to more spurious spikes in the reconstruction, while $\varphi_{GM}$ over-smooths the image and fails to reconstruct its main feature. Just as the results from potential function $\varphi_{HL}$, the self-adaptation of $\delta_n$ in this formulation does not seem to improve the reconstruction significantly, hence a new formulation of the multiplicative model with a different set of recurrence relations is required.



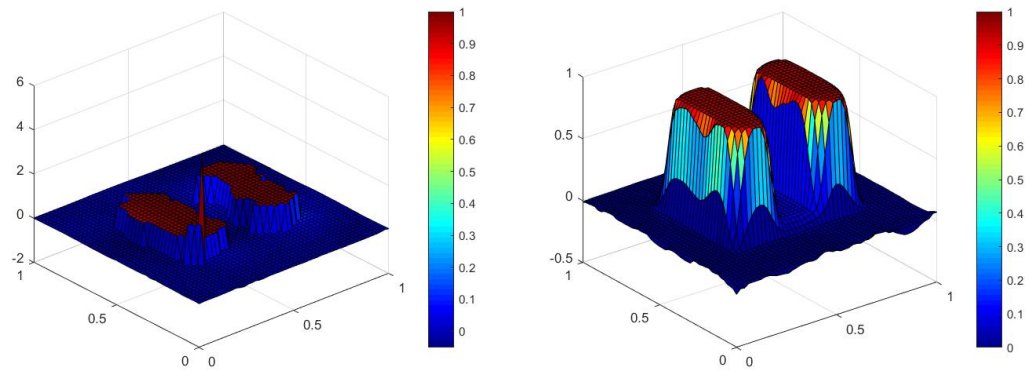
(a) $\varphi_{GM}$ reconstruction (b) $\varphi_{HS}$ reconstruction

Figure 5.4: Reconstruction by MMR with different potential functions (noise implemented: 10% Gaussian)



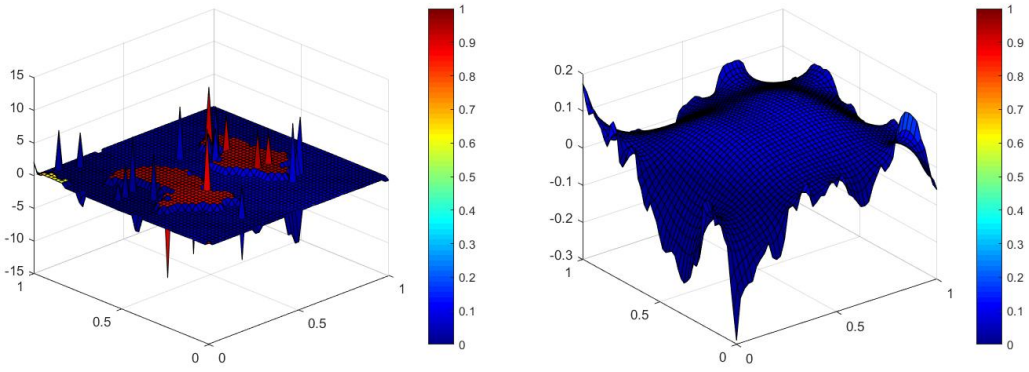
(a) $\varphi_{GM}$ reconstruction (b) $\varphi_{HS}$ reconstruction

Figure 5.5: Reconstruction by MMR with different potential functions (Noise implemented: 50% Gaussian)



(a) $\varphi_{GM}$ reconstruction (b) $\varphi_{HS}$ reconstruction

Figure 5.6: Reconstruction by MSSP with different potential functions (noise implemented: 10% Gaussian)



(a) $\varphi_{GM}$ reconstruction (b) $\varphi_{HS}$ reconstruction

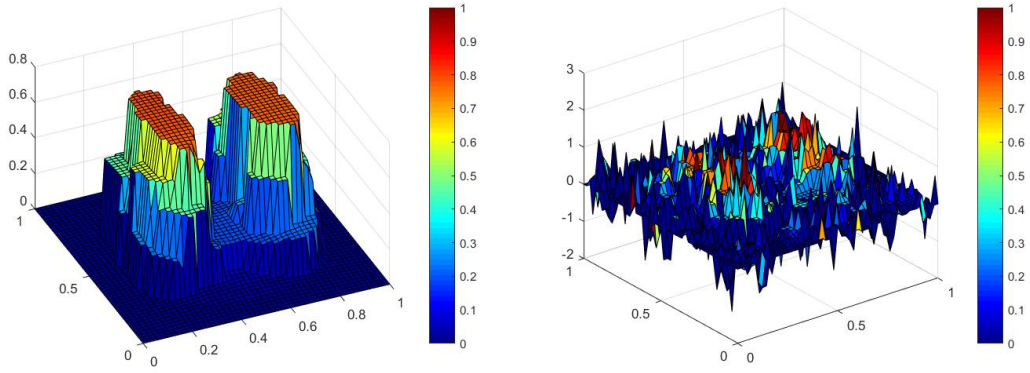
Figure 5.7: Reconstruction by MSSP with different potential functions (noise implemented: 50% Gaussian)

### 5.6.3 Comparison with Additive Counterpart of MMR

In this section I present the numerical results for the additive counterpart of MMR, which is essentially the model by Charbonnier *et al* [CBFAB97] with potential func-

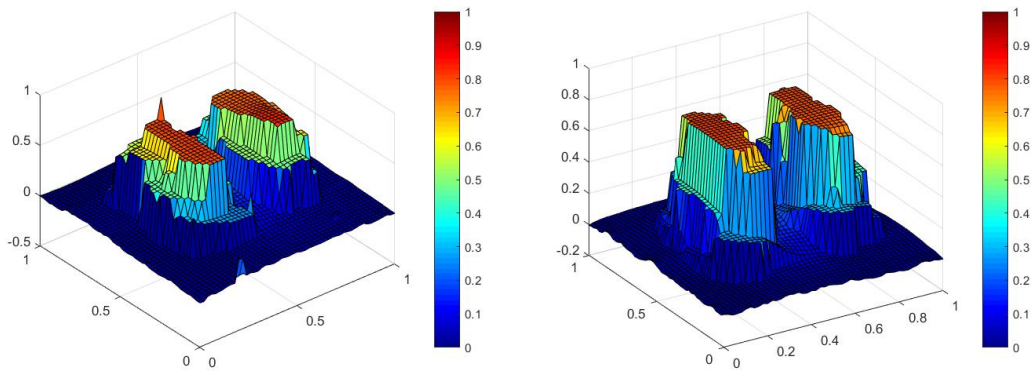
tions $\varphi_{HL}$, $\varphi_{GM}$ and $\varphi_{HS}$ scaled by scaling parameter $\delta$. The test image is again chosen to be the two-block example (1.3.2) with 10% and 50% Gaussian random noise. We see from Fig 5.8 that the additive models are no rivals for their multiplicative counterparts when dealing with high noise amplitude. However, for lower levels of noise we see from Fig 5.9 that the additive model of the $\varphi_{HL}$ type implemented with 10% K-distributed noise appears to be as robust as in the case of 10% Gaussian noise (Fig 5.8 (a)). In other words, the merit of robustness for different noise distributions is shared by both multiplicative and additive models with HL type of potential function.

By comparing Fig 5.10 and Fig 5.4, we observe that the multiplicative models are consistently more robust than their additive counterparts, in terms of different types of potential functions. Generally speaking, the multiplicative models have stronger numerical performance than their additive counterparts, particularly in terms of high noise level tolerance. They also avoid the extra effort of choosing a regularisation parameter $\lambda$, and demonstrate the potential of multiplicative regularisation for further investigation.



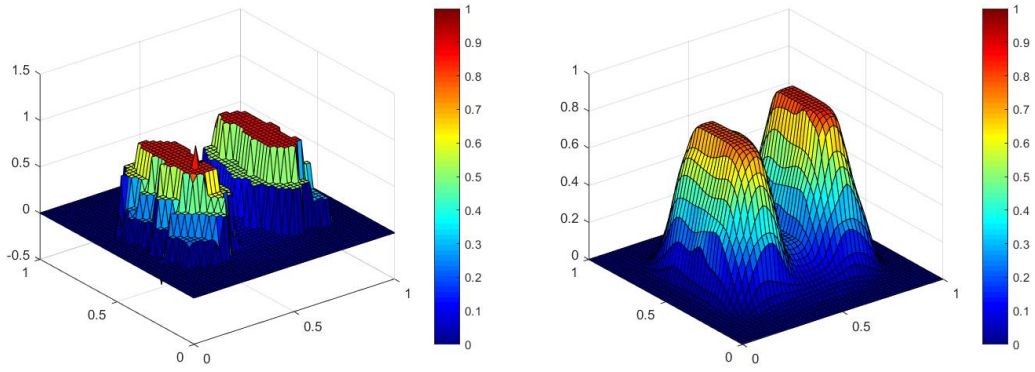
(a) Gaussian 10% reconstruction (b) Gaussian 50% reconstruction

Figure 5.8: Reconstruction by Charbonnier's additive model [CBFAB97] with potential function $\varphi_{HL}$ for 10% and 50% Gaussian noise



(a) K ($\alpha = 0.5$) (spiky),10% , $\lambda = 525$, $\delta^2 = 1$ (b) K ($\alpha = 10$) (asymm bell),10%, $\lambda = 525$, $\delta^2 = 1$

Figure 5.9: Reconstruction by Charbonnier's additive model [CBFAB97] with potential function $\varphi_{HL}$ for 10% K-distributed noise (shape parameter: $\alpha = 0.5$ and $\alpha = 10$)

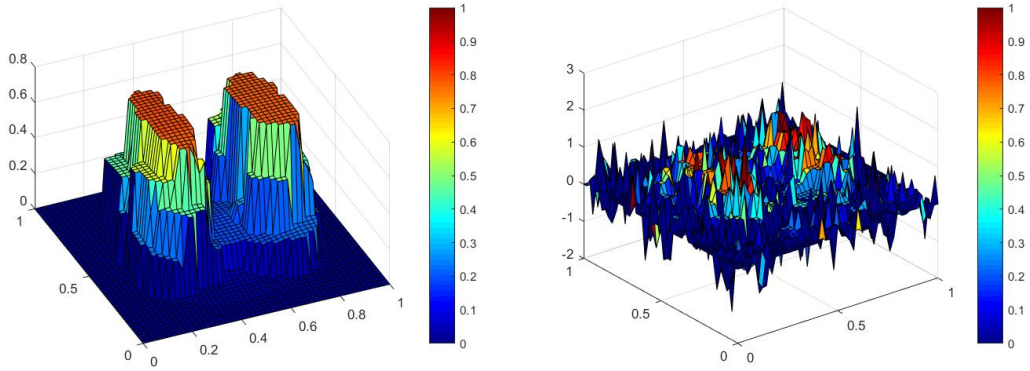


(a) $\varphi_{GM}$ reconstruction, $\lambda = 525$, $\delta^2 = 10$ (b) $\varphi_{HS}$ reconstruction,$\lambda = 1$, $\delta^2 = 0.001$

Figure 5.10: Reconstruction by Charbonnier's additive model [CBFAB97] with potential function $\varphi_{GM}$ and $\varphi_{HS}$ with 10% Gaussian noise implemented

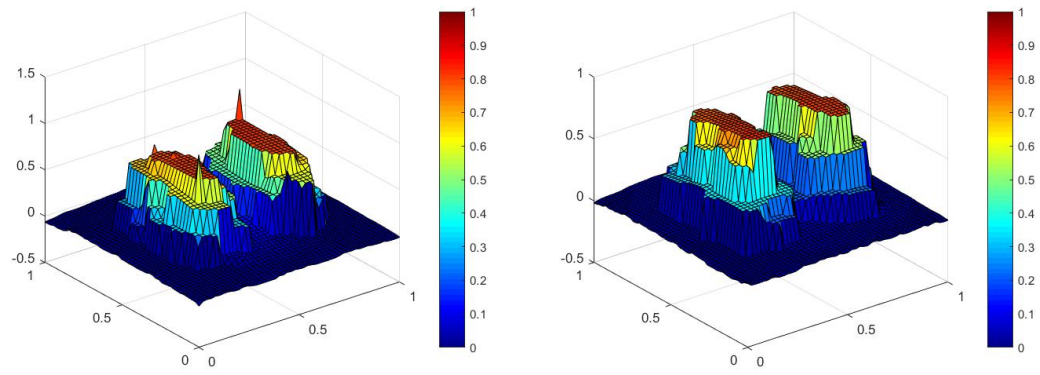
### 5.6.4 Comparison with Additive Counterpart of MSSP

In this section I test the additive model as presented in (5.3.6). This model modifies Charbonnier's model [CBFAB97] by (i) inserting a scaling parameter $\delta$ in the potential function $\psi_{pot}$, and (ii) self-updating both weight function $(b_n)_{i,j}$ and $\delta_n$ in the algorithm via recurrence relations (5.3.4) and (5.3.5). The test image is again chosen to be the two-block example (1.3.2), and the potential functions take the form of $\varphi_{HL}$, $\varphi_{GM}$ and $\varphi_{HS}$ respectively.



(a) Gaussian 10% reconstruction (b) Gaussian 50% reconstruction

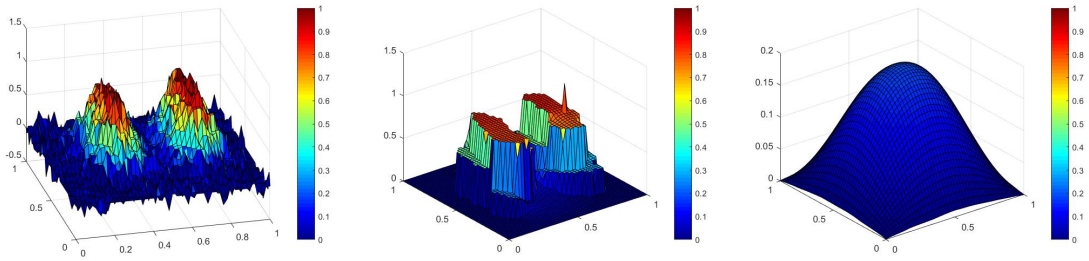
Figure 5.11: Reconstruction by the modified additive model with potential function $\varphi_{HL}$ for 10% and 50% Gaussian noise



(a) K ($\alpha = 0.5$) (spiky),10% , $\lambda = 525$, $\delta^2 = 1$ (b) K ($\alpha = 10$) (asymm bell),10%, $\lambda = 525$, $\delta^2 = 1$

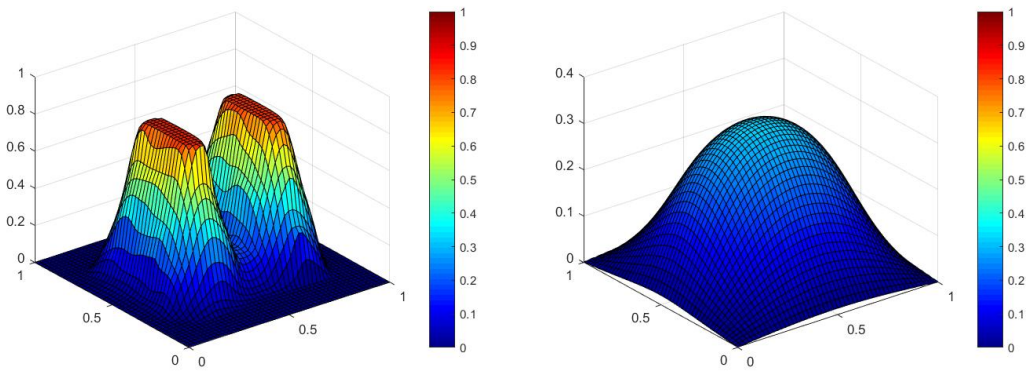
Figure 5.12: Reconstruction by the modified additive model with potential function $\varphi_{HL}$ for 10% K-distributed noise (shape parameter: $\alpha = 0.5$ and $\alpha = 10$)

We observe that the results in Fig 5.11 and Fig 5.12 are at similar level as Fig 5.8 and Fig 5.9 in MMR. It is again verified that despite being robust against different noise distribution, the additive models are no rivals of their multiplicative counterparts when dealing with high noise amplitude.



(a) $\delta_0^2 = 1$ (b) $\delta_0^2 = 10$ (c) $\delta_0^2 = 100$

Figure 5.13: Reconstruction by the modified additive model with potential function $\varphi_{GM}$ with different initial value $\delta_0^2$ (Noise implemented: 10% Gaussian)



(a) $\delta_0^2 = 0.001$ (b) $\delta_0^2 = 1$

Figure 5.14: Reconstruction by the modified additive model with potential function $\varphi_{HS}$ with different initial value $\delta_0^2$ (Noise implemented: 10% Gaussian)

A very interesting phenomenon occurred in these modified additive models: they are more sensitive to the initial value of $\delta_0$ compared to their multiplicative counterparts. Fig 5.13 and Fig 5.14 illustrate this point: it seems that too small or too large a choice of $\delta_0$ will lead to either under-regularised of over-regularised images, and the most appropriate choice of $\delta_0$ agrees with the choice in the additive model with scaling parameter $\delta$ being constant, as demonstrated in Fig 5.10 in MMR. This is likely due to the fact that the additive counterpart of MSSP does not necessarily direct its solution to the global minimum as discussed in Section 5.3.3; non-convexity throughout in the iteration makes this additive model very sensitive to the initial values of the parameters.
Overall, we can conclude the following:

- Multiplicative regularisation models lead to reconstructions superior than their additive counterparts, particularly in the case of highly noisy data, regardless of whether scaling parameter $\delta$ is being self-adaptive, and regardless of the choice of potential function.

- The merit of robustness against different noise distribution is shared by both additive and multiplicative model using HL type of potential function in both

MMR and MSSP, *i.e.* with constant and self-adaptive scaling parameter $\delta$.

- The self-adaptation of this parameter with recurrence relation (5.3.5) does not seem to improve reconstruction significantly in either multiplicative nor additive models. Hence a new set of recurrence relations are called for in order to achieve multiplicative models comparable with the one by Abubakar *et al* [AvdBHB04], which leads to EMM model in the next chapter.

# Chapter 6

# A Novel Enhanced Method

In previous chapters I have presented new multiplicative models obtained by combining features from the multiplicative method by Abubakar *et al* [AvdBHB04] and the additive method by Charbonnier *et al* [CBFAB97]. In these models an extra scaling parameter $\delta$ is introduced in the potential function of the regularisation term: it is firstly kept constant in the MMR method, and later modified to be self-adaptive, and updated via recurrence relations in the MSSP method. Two-step and three-step alternating minimisation algorithms are applied to solve these models iteratively. However, despite admitting more rigorous algorithms and leading to better theoretical results regarding convergence, the numerical performance of both formulations appear to be weaker than Abubakar's multiplicative model [AvdBHB04]. This motivates further research on the self-adaptation of scaling parameter $\delta$. In the spirit of this, in this chapter I will present a further formulation of multiplicative model by replacing the $\delta$ updating rule in MSSP (5.3.5), *i.e.*

$$\sum_{(i,j)} \psi'_{pot}(\delta^2_{n+1}(b^2_{n+1})_{i,j})(b^2_{n+1})_{i,j} = 0,$$

by Abubakar's method [AvdBHB04] in (3.2.3), *i.e.*

$$\delta^2_{n+1} = \frac{\mu}{2}\frac{\sum_{(i,j)}(b^2_{n+1})_{i,j}[(D_x u_n)^2_{i,j} + (D_y u_n)^2_{i,j}]}{\sum_{(i,j)}(b^2_{n+1})_{i,j}},$$

The reason for choosing this recurrence relation is to take advantage of the tuning parameter $\mu$, which was discussed extensively in previous Chapter 4. The parameter selection rule for $\mu$ in this chapter remains the same as shown previously. Detailed derivation of the model will be presented in the next section. In later sections, I will show that the enhanced multiplicative method (EMM) equipped with the same parameter choice rule (4.4.18) leads to numerical results exceeding all other additive and multiplicative models presented previously in Chapter 3, 4 and 5.

## 6.1 Model Derivation

Similarly to previous formulations in this thesis (MMR and MSSP), this model also seeks to modify Abubakar's objective cost functional (3.2.1) by turning it into the form of

$$C(u, b^2, \delta^2) = \{\sum_{(i,j)}[(\mathcal{K}u)_{i,j} - f^{\epsilon}_{i,j}]^2\} \cdot F^{multi}(u, b^2, \delta^2) \tag{6.1.1}$$

where the multiplicative functional takes the form of

$$F^{multi}(u, b^2, \delta^2) = \sum_{(i,j)} b^2_{i,j}[(D_x u)^2_{i,j} + (D_y u)^2_{i,j}] + \sum_{(i,j)} \Phi(b^2_{i,j}, \delta^2) \tag{6.1.2}$$

With the aim of recapturing Abubakar's model good performance, whilst retaining our models' better theoretical results regarding convergence, we now impose that, when the cost functional $C(u, b^2, \delta^2)$ is minimised in both complementary variables

$b^2$ and $\delta^2$, the first order optimality conditions

$$\frac{\partial C(u, b^2, \delta^2)}{\partial (b^2)} = 0, \ and \ \frac{\partial C(u, b^2, \delta^2)}{\partial (\delta^2)} = 0, \tag{6.1.3}$$

will agree with the set of recurrence relations in Abubakar's model [AvdBHB04]

$$(b^2_{n+1})_{i,j} = \frac{1}{[(D_x u_n)^2_{i,j} + (D_y u_n)^2_{i,j} + \delta^2_n]} \ \forall \ i, j = 1, 2, ...N \tag{6.1.4}$$

and

$$\delta^2_{n+1} = \frac{\mu}{2} \frac{\sum_{(i,j)} (b^2_{n+1})_{i,j} [(D_x u_n)^2_{i,j} + (D_y u_n)^2_{i,j}]}{\sum_{(i,j)} (b^2_{n+1})_{i,j}} \tag{6.1.5}$$

Hence the form of potential function $\Phi$ in (6.1.1) is determined by this compatibility accordingly. As before, we consider the minimisation of $C(u, b^2, \delta^2)$ in terms of the complementary variables $\delta^2$ and $b^2_{i,j}$. To simplify notation in the following calculation, we suppress all the subscripts $n$ and $n+1$ in the recurrence relation (6.1.4) and (6.1.5). By substituting the expression of $C(u, b^2, \delta^2)$ in (6.1.1) into the first order optimality conditions (6.1.3), they become

$$[(D_x u)^2_{i,j} + (D_y u)^2_{i,j}] + \frac{\partial \Phi(b^2_{i,j}, \delta^2)}{\partial b^2_{i,j}} = 0, \ \forall \ i, j \in \{1, 2, ...N\} \tag{6.1.6}$$

and

$$\frac{\partial}{\partial \delta^2} \sum_{(i,j)} \Phi(b^2_{i,j}, \delta^2) = 0. \tag{6.1.7}$$

We now need to find such a concrete form for $\Phi$ that equation (6.1.6) and recurrence relation (6.1.4), (6.1.7) and recurrence relation (6.1.5) are satisfied simultaneously. To achieve this goal, we start from the first pair, and try to relate the recurrence

relation (6.1.4) with equation (6.1.6). Re-write (6.1.4) as

$$[(D_x u)^2_{i,j} + (D_y u)^2_{i,j}] + \delta^2 - \frac{1}{b^2_{i,j}} = 0 \ \forall \ i,j = 1,2,...N \tag{6.1.8}$$

Comparing with equation (6.1.6), we conclude that

$$\frac{\partial \Phi(b^2_{i,j}, \delta^2)}{\partial b^2_{i,j}} = \delta^2 - \frac{1}{b^2_{i,j}}, \ \forall \ i,j = 1,2,...N$$

Solving this differential equation in terms of $b^2_{i,j}$ we obtain

$$\Phi(b^2_{i,j}, \delta^2) = -\log b^2_{i,j} + \delta^2 b^2_{i,j} + T(\delta^2), \ \forall \ i,j = 1,2,...N \tag{6.1.9}$$

where $T(\delta^2)$ is some function of $\delta^2$. To further solve for $T(\delta^2)$, we substitute (6.1.9) into equation (6.1.7), and obtain

$$\sum_{(i,j)} b^2_{i,j} + N^2 \frac{\partial}{\partial \delta^2} T(\delta^2) = 0 \tag{6.1.10}$$

Before relating expression (6.1.10) to equation (6.1.5), we firstly substitute equation (6.1.8) into (6.1.5) to cancel the term $[(D_x u)^2_{i,j} + (D_y u)^2_{i,j}]$. This gives rise to an equation involving only $b^2_{i,j}$ and $\delta^2$:

$$\delta^2 \sum_{(i,j)} b^2_{i,j} = \frac{\mu}{2}[-\delta^2 \sum_{(i,j)} b^2_{i,j} + N^2] \tag{6.1.11}$$

By rearranging terms this becomes

$$\sum_{(i,j)} b^2_{i,j} - N^2 \frac{\mu}{\mu + 2} \cdot \frac{1}{\delta^2} = 0 \tag{6.1.12}$$

Equating (6.1.10) and (6.1.12) gives

$$\frac{\partial}{\partial \delta^2} T(\delta^2) = -\frac{\mu}{\mu + 2} \cdot \frac{1}{\delta^2}$$

Solving this differential equation, we obtain

$$T(\delta^2) = -\frac{\mu}{\mu+2}\log\delta^2 + M, \tag{6.1.13}$$

where $M$ is some integration constant to be determined. The appropriate choice for $M$ will be discussed in detail in later sections. For now it is treated as a known parameter in the objective cost functional. We therefore conclude that the explicit form of the potential function $\Phi(b_{i,j}^2, \delta^2)$ is:

$$\begin{aligned}\Phi(b_{i,j}^2, \delta^2) &= -\log b_{i,j}^2 + \delta^2 b_{i,j}^2 - \frac{\mu}{\mu+2}\log\delta^2 + M \\ &= -\log[(\delta^2)^{\frac{\mu}{\mu+2}} b_{i,j}^2] + \delta^2 b_{i,j}^2 + M,\ \forall\ i,j = 1,2,...N,\end{aligned} \tag{6.1.14}$$

which results in an objective cost functional in the form of

$$\begin{aligned}C(u, b^2, \delta^2) &= \{\sum_{(i,j)}[(\mathcal{K}u)_{i,j} - f_{i,j}^{\epsilon}]^2\} \cdot \{\sum_{(i,j)} b_{i,j}^2[(D_x u)_{i,j}^2 + (D_y u)_{i,j}^2] + \delta^2 \sum_{(i,j)} b_{i,j}^2 \\ &\quad - \sum_{(i,j)} \log[(\delta^2)^{\frac{\mu}{\mu+2}} b_{i,j}^2] + N^2 M\}\end{aligned} \tag{6.1.15}$$

**Possible Extension to Other Potential Functions**

I have explored the possibility of generalising the model to potential functions of type $\varphi_{GM}$ and $\varphi_{HS}$. To achieve this goal, I would need to find new forms of the potential function $\Phi(b_{i,j}^2, \delta^2)$ in (6.1.2) such that the first order optimality conditions (6.1.6) and (6.1.7) fit the set of recurrence relations for the weight function $b_{i,j}^2$ appropriate for the potential functions $\varphi_{GM}$ and $\varphi_{HS}$, given Table 5.1, and the new recurrence relation regarding for $\delta^2$ as (6.1.5) for both potential functions $\varphi_{GM}$ and $\varphi_{HS}$. My calculations show that trying to satisfy these conditions leads to potential functions that are essentially of type HL. In other words, the this enhanced multiplicative model (6.1.15) relies on the special form of potential function $\varphi_{HL}$, and it is not possible to generalise it to other types of potential function with an extra parameter $\mu$.

## 6.2 Positivity of Objective Cost Functional $C(u, b^2, \delta^2)$

Similarly to previous formulations, we can again define

$$\begin{aligned}\bar{C}(u,\delta^2) &= \min_{b^2} C(u,b^2,\delta^2) \\ &= \{\sum_{(i,j)}[(\mathcal{K}u)_{i,j} - f^{\epsilon}_{i,j}]^2\} \cdot \{\sum_{(i,j)} \log[(D_x u)^2_{i,j} + (D_y u)^2_{i,j} + \delta^2] \\ &\quad - \sum_{(i,j)} \log[(\delta^2)^{\frac{\mu}{\mu+2}}] + N^2(M+1)\} \end{aligned} \tag{6.2.1}$$

by applying the first recurrence relation (6.1.4) (which is also the first order optimality condition for $b$). We can further define

$$\tilde{C}(u) = \min_{\delta} \bar{C}(u,\delta^2) = \bar{C}(u,\hat{\delta}^2) \tag{6.2.2}$$

where $\hat{\delta}$ satisfies

$$\sum_{(i,j)} \frac{1}{(D_x u)^2_{i,j} + (D_y u)^2_{i,j} + \hat{\delta}^2} - \frac{\mu N^2}{\mu+2} \cdot \hat{\delta}^2 = 0 \tag{6.2.3}$$

To summarise, we construct an objective cost functional $\bar{C}(u,\delta^2)$ in the form of (6.1.15), and minimise it in terms of complementary variables $b$ and $\delta$ and image intensity $u$. This is equivalent to minimising $\bar{C}(u,\hat{\delta}^2)$ in (6.2.2) where $\hat{\delta}^2$ is expressed in terms of $u$ via (6.2.3). It is worth pointing out that the positivity of $\tilde{C}(u) = \bar{C}(u,\hat{\delta}^2)$ in (6.2.2) cannot be guaranteed in general: in the numerical implementation, cases of negative $\bar{C}(u,\hat{\delta}^2)$ may occur. However, we show that by making appropriate choice for the integration factor $M$, the positivity of $\bar{C}(u,\hat{\delta}^2)$ may be enforced.

**Choice of the Integration Factor $M$**

In this section we discuss under what condition the multiplicative regularisation functional $F^{multi}$ in (6.1.2) will be non-negative, and derive a choice rule for the integration factor $M$ in expression. By considering $F^{multi}$ as a function of $\delta^2$ and $b^2_{i,j}$,

we see that the second order derivatives satisfy

$$\frac{\partial^2}{\partial(b_{i,j}^2)^2} F^{multi}(u, b^2, \delta^2) = \frac{1}{(b_{i,j}^2)^2} \geq 0,$$

and

$$\frac{\partial^2}{\partial(\delta^2)^2} F^{multi}(u, b^2, \delta^2) = \frac{\mu}{\mu+2} \frac{1}{(\delta^2)^2} \geq 0.$$

This means that $F^{multi}(u, b^2, \delta^2)$ is convex in variables $\delta^2$ and $b_{i,j}^2$. The first order optimality conditions in these variables $\frac{\partial}{\partial(b_{i,j}^2)} F^{multi}(u, b^2, \delta^2) = 0, \ \forall \ 1 \leq i, j \leq N$ and $\frac{\partial}{\partial(\delta^2)} F^{multi}(u, b^2, \delta^2) = 0$ lead to stationary solutions which correspond to the recurrence relations in Abubakar's model [AvdBHB04]

$$\hat{b}_{i,j}^2 = \frac{1}{(D_x u)_{i,j}^2 + (D_y u)_{i,j}^2 + \hat{\delta}^2} \tag{6.2.4}$$

$$\hat{\delta^2} = \frac{\mu N^2}{(\mu+2) \sum_{i,j} \hat{b}_{i,j}^2} \tag{6.2.5}$$

Note that in these stationary equations, the solution $\hat{\delta}^2$ and $\hat{b}_{i,j}^2, \ 1 \leq i, j \leq N$ still have dependence on the image intensity $u$. They should be interpreted as implicit functions of $u$. Here equations (6.2.4) and (6.2.5) are derived directly from the recurrence relations regarding complementary variables $b_n$ and $\delta_n$ which are enforced throughout the algorithm for each image intensity $u_n$, and no convergence of $u_n$ is assumed. We use $F^{multi}(u, b^2, \delta^2)$ to denote the multiplicative regularisation functional with arbitrary choice of $u$, $b$ and $\delta$. It may be bounded below by taking $b^2 = \hat{b}^2$, $\delta^2 = \hat{\delta}^2$ in (6.1.2):

$$F^{multi}(u, b^2, \delta^2) \geq F^{multi}(u, \hat{b}^2, \hat{\delta}^2) = N^2(M+1) - \sum_{(i,j)} \log[\hat{b}_{i,j}^2] - \frac{\mu N^2}{\mu+2} \log(\hat{\delta}^2), \tag{6.2.6}$$

since $\sum_{(i,j)} \hat{b}^2_{i,j}[(D_x u)^2_{i,j} + (D_y u)^2_{i,j} + \hat{\delta}^2] = N^2$. By using (6.2.5) to write

$$\sum_{(i,j)} \hat{b}^2_{i,j} = \frac{\mu N^2}{\mu + 2} \cdot \frac{1}{\hat{\delta}^2},$$

we obtain

$$\begin{aligned} -\sum_{(i,j)} \log(\hat{b}^2_{i,j}) &= -\log\left(\prod_{i,j} \hat{b}^2_{i,j}\right) \geq -\log\left[\left(\frac{\sum_{i,j} \hat{b}^2_{i,j}}{N^2}\right)^{N^2}\right] \\ &= -N^2 \log\left(\frac{\mu}{\mu+2}\right) + N^2 \log(\hat{\delta}^2). \end{aligned} \tag{6.2.7}$$

Now, using (6.2.7) in (6.2.6) we obtain the inequality

$$F^{multi}(u, b^2, \delta^2) \geq N^2(M+1) - N^2 \log(\frac{\mu}{\mu+2}) + \frac{2}{\mu+2} N^2 \log(\hat{\delta}^2),$$

and, in order to ensure the positivity of $F^{multi}(u, b, \delta^2)$, the following criterion is sufficient:

$$N^2(M+1) - N^2 \log(\frac{\mu}{\mu+2}) + \frac{2}{\mu+2} N^2 \log(\hat{\delta}^2) \geq 0.$$

This is equivalent to

$$\hat{\delta}^2 \geq (\frac{\mu}{\mu+2})^{\frac{\mu+2}{2}} \exp\{-\frac{(M+1)(\mu+2)}{2}\}. \tag{6.2.8}$$

Recall that $\hat{\delta}^2$ is an implicit function of image intensity $u$ by definition. This means that in order to ensure positivity of $F^{multi}(u, b, \delta^2)$, the criterion (6.2.8) needs to be enforced for all possible values of $u$.

We observe that the first factor on the right hand side of 6.2.8 is always less than 1. Here $\mu$ should be chosen according to criterion 4.4.18 derived previously, as confirmed in Section 6.4 below, and numerical implementations normally suggest a value in the

region of 10.

The integration factor $M$ could, in principle, be chosen arbitrarily large, to make the whole right hand side as close as we like to zero. In practice, we found that the numerical reconstruction is not very sensitive to the choice of $M$, and $M \in (-1, 3]$ is enough to enforce positivity of the cost functional, and leads to good reconstruction in most applications.

It may be observed from the structure of (6.1.15) that apart from keeping $C(u, b^2, \delta^2)$ positive, $M$ is also responsible for balancing the fidelity term $F(u)$ in the objective cost functional $C(u, b^2, \delta^2)$. In other words, it plays the role of an implicit regularisation parameter. For very large $M$, then, minimising $C(u, b^2, \delta^2)$ is almost equivalent to minimising the data fidelity $F(u)$, which corresponds to the direct inversion of the original inverse problem. With the insertion of the parameter $M$, the model can be considered as a mixture of additive and multiplicative regularisation, which may explain its superior numerical results over all other additive and multiplicative models presented previously. This observation also leads us to conclude that one should choose a large value of $M$ for small noise, and a small value of $M$ for large noise.

It is worth noticing that in the case of $\mu = 10$, the choice of $M = 3$ result in a bound of $\hat{\delta}^2(u) \geq \mathcal{O}(10^{-11})$ according to criterion (6.2.8). This is already an extremely small value, which means the positivity of objective cost functional $C(u, b, \delta)$ may be enforced in most applications.

## 6.3 Convergence

### 6.3.1 Three-Step Modified Alternating Minimisation

Similarly to MSSP, we will again apply the modified alternating minimisation algorithm (Algorithm 5.3.1) to minimise the objective functional (6.1.15) in terms of

the complementary variables $b$, $\delta$ and the image intensity $u$ alternately. This is essentially the same as Algorithm 5.3.1, but for the new cost functional (6.1.15) with different updating rules for the variables:

**Algorithm 6.3.1.** ***Three-step modified alternating minimisation algorithm***
*Initialization:* $u_0^{M_0} = f^\epsilon$, $\delta_0$, $M_{inner}$ *and* $M_{outer}$ *chosen as in Algorithm 5.2.1 and Algorithm 5.3.1;*
*For* $0 \leq n \leq M_{outer}$, *Repeat*
***Outer loop:***

- *Fix* $\delta = \delta_n$ *and update* $b_{n+1}$ *according to (6.1.4);*
- *Fix* $b = b_{n+1}$ *and update* $\delta_{n+1}$ *according to (6.1.5);*

***Inner loop:*** *Fix* $b = b_{n+1}$ *and* $\delta = \delta_{n+1}$ *and iterate for* $u$ *on the superscript* $m$
*Initialization:* $u_{n+1}^0 = u_n^{M_n}$.
*If* $1 \leq m \leq M_{inner}$, *then*

- *Update image intensity* $u_{n+1}^{m+1}$ *as in Algorithm 5.2.1 and Algorithm 5.3.1, but replacing the regularisation term in the old formulation by the new one in (6.1.15);*

*The termination criteria for both the inner and outer loops are the same as in Algorithm 5.2.1 and Algorithm 5.3.1.*

### 6.3.2 Convergence Property of $C(u_n, b_n^2, \delta_n^2)$

To discuss the convergence property of the three-step minimisation algorithm presented in the previous section, we will start from the convergence of the objective cost functional $C(u_n, b_n^2, \delta_n^2)$. Its monotonic decrease may be verified easily, as the algorithm minimises in the three variables $b^2$, $\delta^2$ and $u$ alternately: $(b_{n+1}^2)_{i,j}$ and $\delta_{n+1}^2$ are taken to be the unique minimiser $\arg\min_b C(u_n, b^2, \delta_n^2)$ and $\arg\min_\delta C(u_n, b_{n+1}^2, \delta^2)$ due to the convexity of functional $C(u, b^2, \delta^2)$ in (6.1.15) in these complementary variables, while $u_{n+1}$ is guaranteed to be one of the critical points of $C(u, b_{n+1}^2, \delta_{n+1}^2)$

via gradient descent method applied in the inner loop with the appropriate choice of step size (as shown in Proposition 5.2.2).

The only difference in EMM is that we first have to ensure non-negativity of $C(u, b^2, \delta^2)$ in (6.1.15) by choosing the integration factor $M$ appropriately, as shown in Section 6.2. Once the non-negativity of $C(u, b^2, \delta^2)$ is fulfilled, the monotonically decreasing sequence $C(u_n, b_n^2, \delta_n^2)$ will be bounded below by 0, and its convergence follows naturally. This will be assumed throughout this thesis.

### 6.3.3 Asymptotic Properties of Weight Function

In this section I will show convergence of $[(b_n^2)_{i,j} - (b_{n+1}^2)_{i,j}]^2 \to 0$, with the following assumptions:
(a) Objective cost functional $C(u_n, b_n^2, \delta_n^2)$ is bounded below by 0;
(b) $u_n$ is bounded away from $(\mathcal{K}^*\mathcal{K})^{-1}\mathcal{K}^* f^\epsilon$, *i.e.*, $\exists \Lambda > 0$ such that

$$\sum_{(i,j)} [(\mathcal{K}u)_{i,j} - f_{i,j}^\epsilon]^2 \geq \Lambda, \ n \to \infty.$$

Similarly to what we showed for the previous models MMR and MSSP, Taylor expansion of $C(u_n, b^2, \delta_n^2)$ in (6.1.15) at $b^2 = b_{n+1}^2$ leads to:

$$\begin{aligned} & C(u_n, b_n^2, \delta_n^2) - C(u_n, b_{n+1}^2, \delta_n^2) \\ = \ & \frac{1}{2}\{\sum_{(i,j)} [(\mathcal{K}u)_{i,j} - f_{i,j}^\epsilon]^2\} \cdot \sum_{(i,j)} \psi''_{HL}(c_{i,j}^2)[(b_n^2)_{i,j} - (b_{n+1}^2)_{i,j}]^2, \end{aligned} \tag{6.3.1}$$

where $c_{i,j}^2 \in [(b_{n+1}^2)_{i,j}, (b_n^2)_{i,j}]$, and the potential function takes the HL form

$$\psi_{HL}(\omega) = \omega - \log(\omega) - 1, \quad \psi''_{HL}(\omega) = \frac{1}{\omega^2}.$$

As discussed in Chapter 5.5.2, this special form of HL type by Hebert and Leahy [LH89] leads to the separation of complementary variables $b$ and $\delta$. With assumption

(b), it is hence possible to find a lower bound $\psi''_{HL}(c^2_{i,j}) \geq D_1$ for some $D_1 > 0$, by Charbonnier's argument [CBFAB97] presented in Appendix C (ii) of this dissertation. This will force $[(b^2_n)_{i,j} - (b^2_{n+1})_{i,j}]^2 \to 0$ as $n \to \infty$ in (6.3.1).

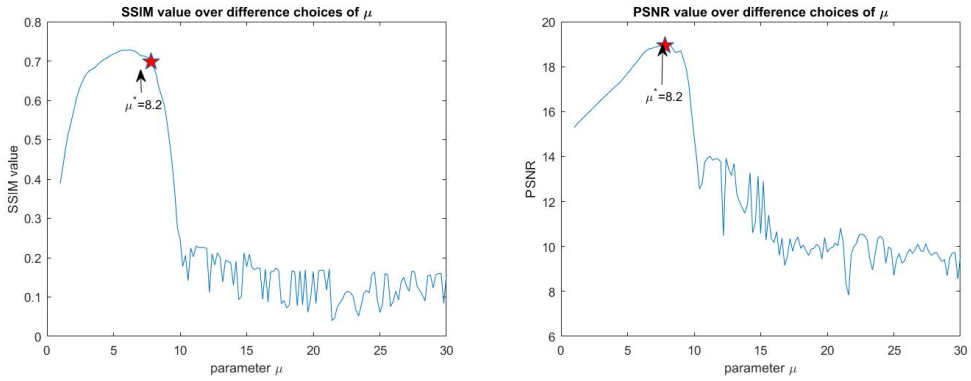
## 6.4 Choice of Parameter $\mu$

As parameter $\delta$ is kept adaptive in this formulation of the multiplicative model with limited theoretical guarantee in its convergence, the tuning of parameter $\mu$ again becomes crucial for achieving good reconstruction. A natural choice rule for $\mu$ is the formula (4.4.18) derived in Chapter 4

$$\mu = \mu^* = 2\sqrt{\frac{1-c}{c}}.$$

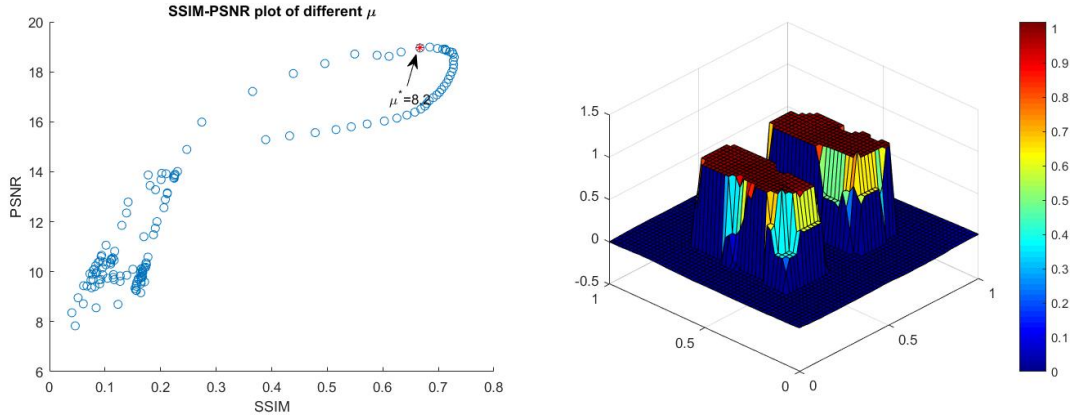
As this selection rule is derived from the recurrence relations of Abubakar's model [AvdBHB04] which is also used in EMM, it is expected to be valid and effective, at least in theory, in this new model as well, as the derivation in Chapter 4 relies only on the set of recurrence relations employed. To numerically justify this formula, I have tested it on the two-block test image (1.3.2) with mesh size $h = 0.02$ and Gaussian 10% noise. According to Table 4.1, the formula predicts $\mu^* = 8.2$. Figure 6.1 shows the SSIM and PSNR index of the reconstructed images over different choices of $\mu$ respectively, with the choice $\mu^* = 8.2$ marked in red. These plots demonstrate the power of formula (4.4.18) in the EMM model. The scatter plot and the actual reconstruction presented in Fig 6.2 reinforce this conclusion.

Similarly to Abubakar's model [AvdBHB04], EMM inherits the "stair-casing" effect as discussed in Section 4.5.3 when dealing with more general gray-level images. This is reflected in the choice between $\mu^*$ and $\mu^*_{cor}$ for images with different feature and noise level. In the numerical implementation, we will choose its value according to Table 4.9.




(a) SSIM plot (b) PSNR plot

Figure 6.1: SSIM and PSNR index of the reconstructed images over different choices of $\mu$ in the case of 10% Gaussian noise for the two-block test image (1.3.2), (Corrupted image: SSIM=0.04, PSNR=13.16)




(a) Scatter plot of PSNR-SSIM index (b) Reconstruction, SSIM=0.64, PSNR=18.95

Figure 6.2: Scatter plot of PSNR-SSIM index of the reconstructed image over different choices of $\mu$, and the actual reconstruction at $\mu = \mu^* = 8.2$

## 6.5 Comparison with Abubakar's Model

In this section, we compare the multiplicative model by Abubakar *et al* [AvdBHB04] to the other multiplicative models MMR, MSSP and EMM. Recall from (3.2.1) in Chapter 3, the objective cost functional in Abubakar's model is given by the following function of $u$, $b^2$ and $\delta^2$

$$C^{Abubakar}(u, b^2, \delta^2) = \{\sum_{(i,j)}[(\mathcal{K}u)_{i,j} - f^{\epsilon}_{i,j}]^2\} \cdot \{\sum_{(i,j)} b^2_{i,j}[(D_x u)^2_{i,j} + (D_y u)^2_{i,j}] + \sum_{(i,j)} \delta^2 b^2_{i,j}\} \tag{6.5.1}$$

where we have suppressed all the iteration indices to simplify notation. Unlike MMR, MSSP and EMM, neither of the two recurrence relations (3.2.2) and (3.2.3) updates $b^2_{n+1}$ and $\delta^2_{n+1}$ following

$$b^2_{n+1} = \min_{b^2} C^{Abubakar}(u_n, b^2, \delta^2_n);$$
$$\delta^2_{n+1} = \min_{\delta^2} C^{Abubakar}(u_n, b^2_{n+1}, \delta^2).$$

This means that the objective cost functional (6.5.1) is not being minimised in the algorithm, hence monotonic descent of the chain of iterated cost functionals, like (5.4.5) cannot be maintained for this model. This may explain why there are few theoretical results about it, and possibly leads to the instability of the weight function $b^2_n$ as was shown in Fig 4.2 of Chapter 4. In order to shed some light on the theoretical properties of the cost functional and the iteration process in Abubakar's method, I sought to write its cost functional in terms of the same type of potential functions as those in Charbonnier *et al* [CBFAB97], derive the complementary variables by minimisation. Noting that, the cost functional (6.5.1) can be written in terms of a potential function $\psi_{\delta^2}$

$$\psi_{\delta^2}(\omega) = \frac{\delta^2}{2}\omega, \tag{6.5.2}$$

we can seek a corresponding potential function $\varphi_{\delta^2}$ such that

$$\min_{b^2} C^{Abubakar}(u, b^2, \delta^2) = \bar{C}^{Abubakar}(u, \delta^2), \tag{6.5.3}$$

and

$$\bar{C}^{Abubakar}(u, \delta^2) = \{\sum_{(i,j)}[(\mathcal{K}u)_{i,j} - f^{\epsilon}_{i,j}]^2\} \cdot \{\sum_{(i,j)} \varphi_{\delta^2}\big(\sqrt{(D_x u)^2_{i,j} + (D_y u)^2_{i,j}}\big)\}. \tag{6.5.4}$$

The construction follows Theorem 2.5.1 and Remark 2.5.2 as in the Charbonnier model [CBFAB97] discussed in Chapter 2. The calculation below is guided by Remark 2.5.2. The potential functions $\psi_{\delta^2}$ and $\varphi_{\delta^2}$ satisfy

$$\begin{aligned} \psi_{\delta^2}(\omega) = \theta_{\delta^2}[(\theta'_{\delta^2})^{-1}(\omega)] - \omega \cdot (\theta'_{\delta^2})^{-1}(\omega), \\ \theta_{\delta^2}(t) = \varphi_{\delta^2}(\sqrt{t}). \end{aligned} \tag{6.5.5}$$

To simplify notation, let

$$a = \frac{\delta^2}{2} > 0,$$

and solving for $\theta_{\delta^2}(\omega)$ leads to

$$a\omega = \theta_{\delta^2}[(\theta'_{\delta^2})^{-1}(\omega)] - \omega \cdot (\theta'_{\delta^2})^{-1}(\omega), \ \textit{for some } a > 0 \tag{6.5.6}$$

Using the change variable: $x = (\theta'_{\delta^2})^{-1}(\omega)$, so $\theta'_{\delta^2}(x) = \omega$, in equation (6.5.6), we get

$$a\theta'_{\delta^2}(x) = \theta_{\delta^2}(x) - x\theta'_{\delta^2}(x)$$

Note that this is a separable differential equation

$$\frac{\theta'_{\delta^2}(x)}{\theta_{\delta^2}(x)} = \frac{1}{x + a},$$

so integrate on both sides gives

$$\log \theta_{\delta^2}(x) = \log(x + a) + const,$$

Hence

$$\begin{aligned} \theta_{\delta^2}(x) &= A(x + a), \\ \varphi_{\delta^2}(t) = \theta_{\delta^2}(t^2) &= At^2 + a \end{aligned} \tag{6.5.7}$$

for some positive constant $A > 0$ to be determined.

So far everything seems to be nice and simple: we have just managed to obtain a closed form expression for the potential function $\varphi_{\delta^2}(t)$, which relates to $\psi_{\delta^2}(\omega)$ in Abubakar's multiplicative model. However, is this expression (6.5.7) really what we expected?

Firstly we notice that $\varphi_{\delta^2}(0) = a = \frac{\delta^2}{2} \neq 0$, so we cannot in any straightforward way use the theorems for Chapter 2 to derive convergence results. Secondly, we see from (6.5.7) that

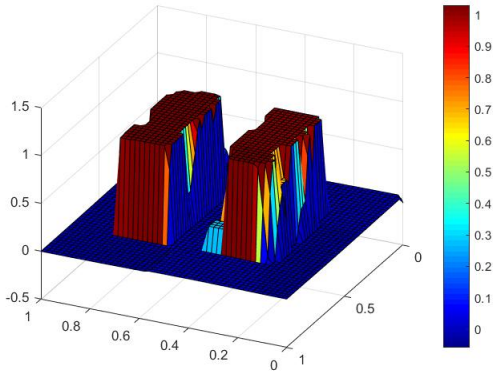
$$\begin{aligned} \theta_{\delta^2}(t) = \varphi_{\delta^2}(\sqrt{t}) &= At + a, \\ \theta'_{\delta^2}(t) &= A, \end{aligned} \tag{6.5.8}$$

so there is no inverse function for $\theta'_{\delta^2}$, which is needed in (6.5.5) for constructing $\psi_{\delta^2}$. Therefore it will never be possible to construct a $\psi_{\delta^2}$ function in the cost functional $C^{Abubakar}(u, b^2, \delta^2)$ using expression of $\varphi_{\delta^2}(t)$ as obtained in (6.5.7). This violation is fatal to this approach, and possibly leads to the instability of the weight function $b_n^2$ in Abubakar's original formulation. It is particularly this instability that prompted us to modify Abubakar's original objective cost functional (6.5.1), and therefore create new multiplicative models such as MMR, MSSP and EMM as discussed in this dissertation.

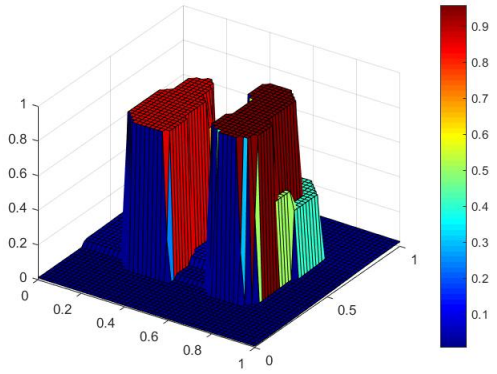
## 6.6 Numerical Results

### 6.6.1 Test on Two-block Image

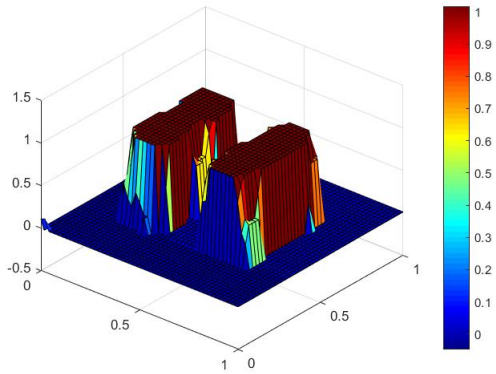
To test the numerical performance of this new EMM, firstly I compare it with Abubakar's model [AvdBHB04] using the two-block test image (1.3.2) for different noise amplitude and distribution. All results are obtained with $M = 3$ and $\mu$ chosen according to parameter choice (4.4.18). Quantitative and qualitative results are presented in Table 6.1 and Fig 6.3. It is apparent that this new formulation is superior not just to Abubakar's model, but also to both MMR and MSSP: it exceeds Abubakar's model consistently for all noise distributions in terms of both image metrics SSIM and PSNR indices, as shown in Table 6.1, and has better qualitative behaviour, as can be seen by comparing Fig 6.3 (EMM reconstruction) with Fig 5.2 (MMR) and Fig 5.3 (MSSP), and with Fig 3.4 and 3.6 (Abubakar's model). Further, results in Fig 6.4 demonstrate the efficiency of EMM: comparison shows that EMM is superior numerically to Abubakar's model [AvdBHB04] in terms of both efficiency and stability, as measured by the SSIM and PSNR indices. We can see that both indices for EMM reach their peak value very quickly (in 100-200 iterations), and remain roughly constant for up to 2000 iterations after that. For Abubakar's model, on the other hand, the SSIM oscillates markedly before reaching its peak at about 500 iterations, and drops appreciably after that; the PSNR shows fewer oscillations and reaches its peak much more quickly (in approximately 50 iterations), but drops sharply after that, making best reconstruction very sensitive to small variations in the number of iterations.



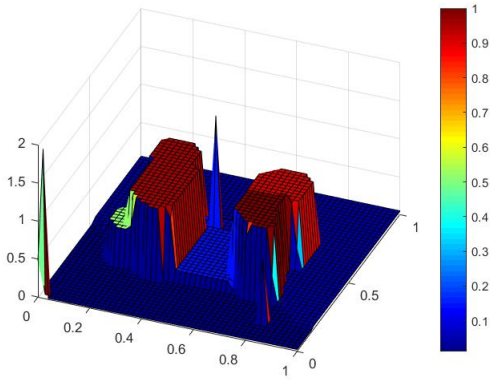
(a) Gaussian 10%



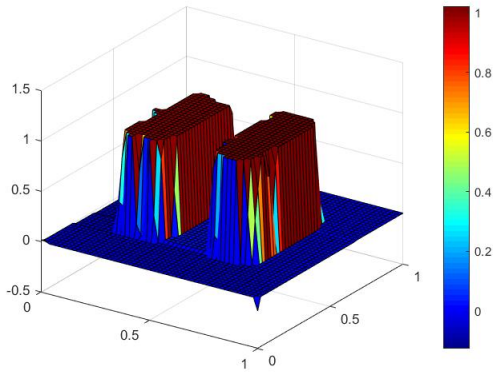
(b) Gaussian 50%



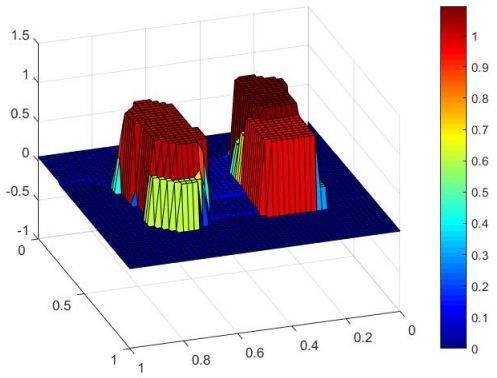
(c) K ($\alpha = 0.5$) (spiky) 10%



(d) K ($\alpha = 0.5$) (spiky) 50%

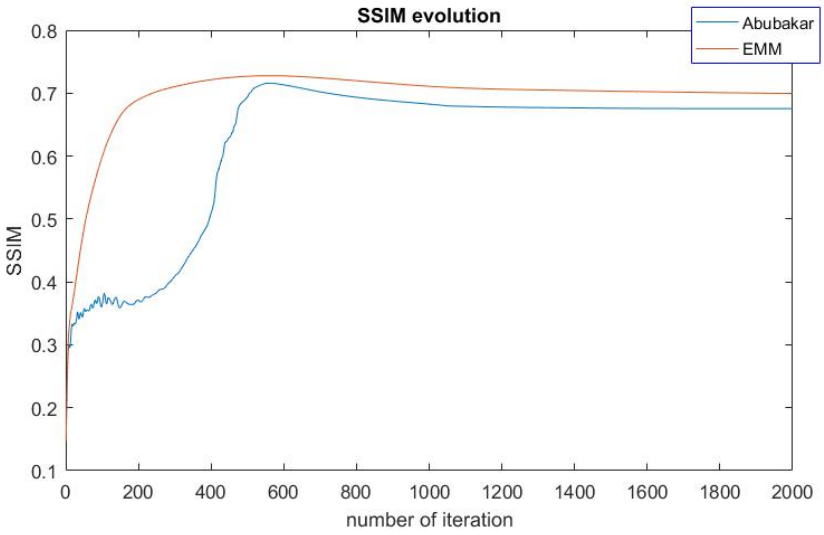


(e) K ($\alpha = 10$)(asymm bell) 10%

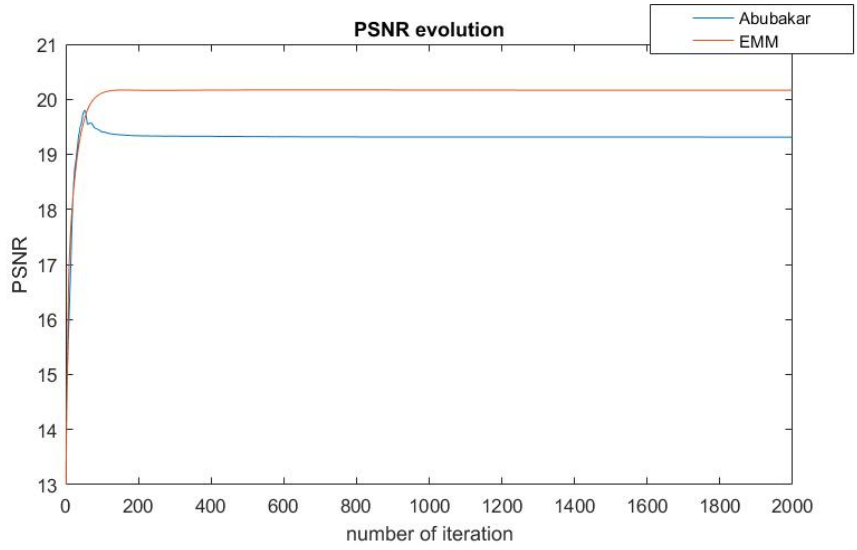


(f) K ($\alpha = 10$)(asymm bell)

Figure 6.3: Reconstructed images by the EMM model for various noise distributions at 10% and 50% amplitude




(a) SSIM evolution




(b) PSNR evolution

Figure 6.4: SSIM and PSNR evolution of reconstruction $u_n$, for both Abubakar's model and the EMM model (noise implemented: 10% Gaussian)

| Noise type | Model by | SSIM | PSNR |
|---|---|---|---|
| Gaussian | Abubakar | 0.68 | 19.38 |
| Gaussian | EMM | 0.72 | 20.17 |
| K ($\alpha = 10$) (asymm bell) | Abubakar | 0.71 | 19.54 |
| K ($\alpha = 10$) (asymm bell) | EMM | 0.74 | 21.82 |
| K ($\alpha = 0.5$)(spiky) | Abubakar | 0.70 | 19.64 |
| K ($\alpha = 0.5$)(spiky) | EMM | 0.77 | 22.07 |

Table 6.1: SSIM and PSNR of the Reconstruction by Abubakar and the enhanced multiplicative model EMM (All noise implemented are at 10% level)

### 6.6.2 Test on Shepp-Logan Phantom

In this section I compare all new multiplicative models constructed and implemented in this thesis (of the HL type of potential function) with Abubakar's model [AvdBHB04], using the Shepp-Logan Phantom as a test image. Similarly to the previous two-block test image, all the results in this section are obtained with $M = 3$ and $\mu$ chosen according to parameter choice (4.4.18). In terms of quantitative behaviours, *i.e.* the image metric SSIM and PSNR index, the performance ranking is as follows: EMM > Abubakar's Model > MMR and MSSP, as shown in Table 6.2 and Table 6.3. Here the Shepp-Logan Phantom I used is 256*256 pixels, and the "corrupted" images are firstly blurred with a uniform 9*9 kernel before being polluted by noise with different distributions and amplitudes.

| | Gaussian | K ($\alpha = 10$) (asymm bell) | K ($\alpha = 0.5$)(spiky) |
|---|---|---|---|
| Corrupted | 0.08 | 0.09 | 0.09 |
| Abubakar | 0.79 | 0.80 | 0.77 |
| MMR(HL) | 0.78 | 0.74 | 0.71 |
| MSSP(HL) | 0.78 | 0.78 | 0.71 |
| EMM | 0.82 | 0.85 | 0.83 |

Table 6.2: SSIM index of reconstructed Shepp-Logan Phantom using different multiplicative models with various noise distributions (All noise implemented at 10% level)

| | Gaussian | K ($\alpha = 10$) (asymm bell) | K ($\alpha = 0.5$)(spiky) |
|---|---|---|---|
| Corrupted | 16.32 | 16.31 | 16.26 |
| Abubakar | 19.27 | 19.22 | 19.16 |
| MMR(HL) | 19.13 | 19.01 | 18.77 |
| MSSP(HL) | 19.12 | 18.92 | 18.85 |
| EMM | 20.92 | 20.94 | 20.76 |

Table 6.3: PSNR index of reconstructed Shepp-Logan Phantom using different multiplicative models with various noise distributions (All noise implemented at 10% level)

The qualitative results showing the reconstructed image of the Shepp-Logan Phantom corrupted with 10% noise with various distributions are presented in Fig 6.5 - Fig 6.8. It may be observed that at this amplitude, the qualitative behaviour of different models differs slightly. Abubakar's model and the EMM appear to recover the edges better than the other two models with smoother curves, and all the models are robust against different noise distributions.

(a) Gaussian (b) K($\alpha = 10$,asymmetric bell) (c) K($\alpha = 0.5$, spiky)

Figure 6.5: Reconstructed images by Abubakar's model from 10% noise of various distributions

(a) Gaussian (b) K($\alpha = 10$,asymmetric bell) (c) K($\alpha = 0.5$, spiky)

Figure 6.6: Reconstructed images by MMR from 10% noise of various distributions

(a) Gaussian (b) K($\alpha = 10$,asymmetric bell) (c) K($\alpha = 0.5$, spiky)

Figure 6.7: Reconstructed images by MSSP from 10% noise of various distributions

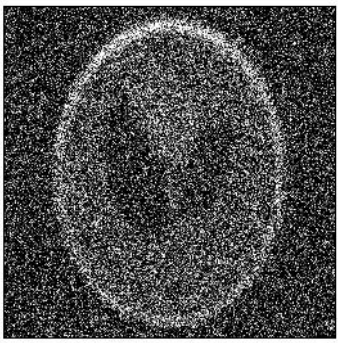
(a) Gaussian (b) K($\alpha = 10$,asymmetric bell) (c) K($\alpha = 0.5$, spiky)

Figure 6.8: Reconstructed images by EMM from 10% noise of various distributions

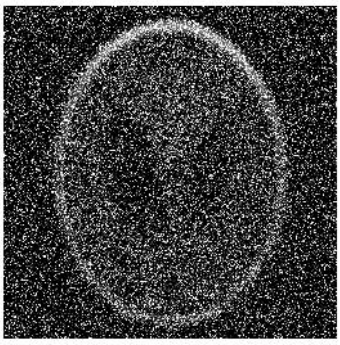
However, for high noise level up to 50%, only Abubakar's model [AvdBHB04] and the EMM are able to cope. Quantitative and qualitative results are presented in Table 6.4 and Fig 6.10 and Fig 6.11 for different noise distribution. It may be observed that the EMM not only leads to better reconstruction consistently in terms of different image metrics, but is also able to extract finer scaled features than Abubakar's model [AvdBHB04] at this noise amplitude. In the Abubakar reconstructions of Fig 6.10 (a) and (b), one of the inclusion is completely missing, while they are still recovered by EMM in both Fig 6.11 (a) and (b). This provides strong evidence that the EMM is more powerful than all multiplicative models presented previously.

| Noise Type | Model | SSIM | PSNR |
|---|---|---|---|
| Gaussian | Corrupted | 0.01 | 5.82 |
| Gaussian | Abubakar | 0.61 | 17.43 |
| Gaussian | EMM | 0.61 | 17.86 |
| K ($\alpha = 10$) (asymm bell) | Corrupted | 0.01 | 5.77 |
| K ($\alpha = 10$) (asymm bell) | Abubakar | 0.57 | 17.16 |
| K ($\alpha = 10$) (asymm bell) | EMM | 0.61 | 17.95 |
| K ($\alpha = 0.5$)(spiky) | Corrupted | 0.01 | 5.75 |
| K ($\alpha = 0.5$)(spiky) | Abubakar | 0.41 | 14.79 |
| K ($\alpha = 0.5$)(spiky) | EMM | 0.54 | 16.82 |

Table 6.4: SSIM and PSNR index of reconstructed Shepp-Logan Phantom using Abubakar's multiplicative model and EMM with high noise amplitude at 50% level.



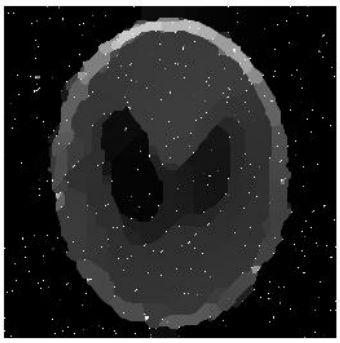
(a) Gaussian



(b) K ($\alpha = 0.5$)(spiky)

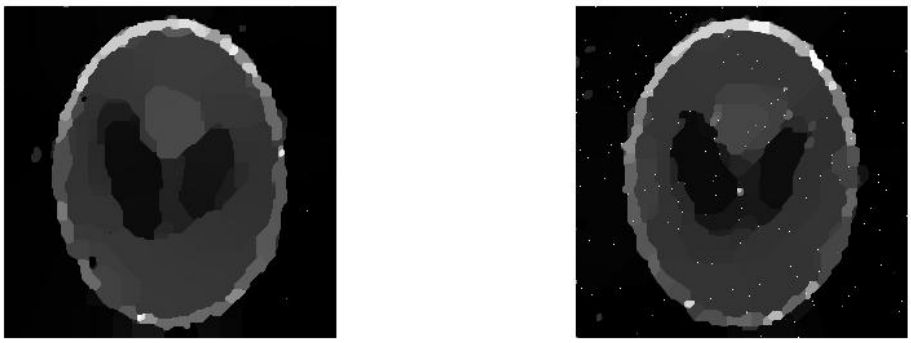
Figure 6.9: Corrupted images at 50% noise level of different distributions

(a) Gaussian



(b) K ($\alpha = 0.5$)(spiky)

Figure 6.10: Reconstructed images from 50% noise level of different distributions using Abubakar's model



(a) Gaussian (b) K ($\alpha = 0.5$)(spiky)

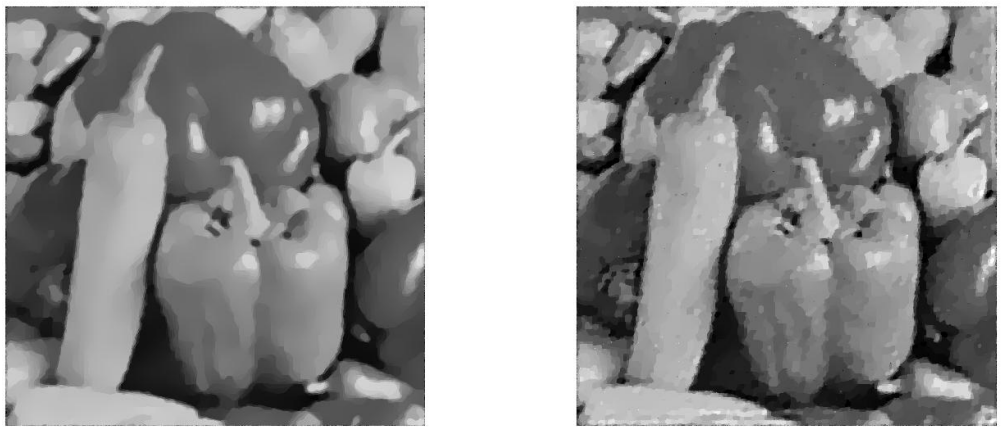
Figure 6.11: Reconstructed images from 50% noise level of different distributions using EMM model

### 6.6.3 Test on Gray-level Digital Images

In this section I compare EMM to Abubakar's method for reconstruction of the gray-level digital image "Peppers" using the same choice of tuning parameter $\mu$. The digital image I used here has 512*512 pixel, and the "corrupted" image are firstly blurred with a uniform 9*9 kernel before being polluted by noise with different distributions and amplitudes. Similarly to Abubakar's method as discussed in Section 4.5 of this dissertation, $\mu^*_{cor}$ derived from the corrupted image outperforms $\mu^*$ due to the stair-casing effect at relative low noise level, while at high noise level, $\mu^*$ appears to be more effective. Numerical results show that EMM performs equally well as Abubakar's method for general digital images, and is robust against different noise distribution and effective at high noise level.

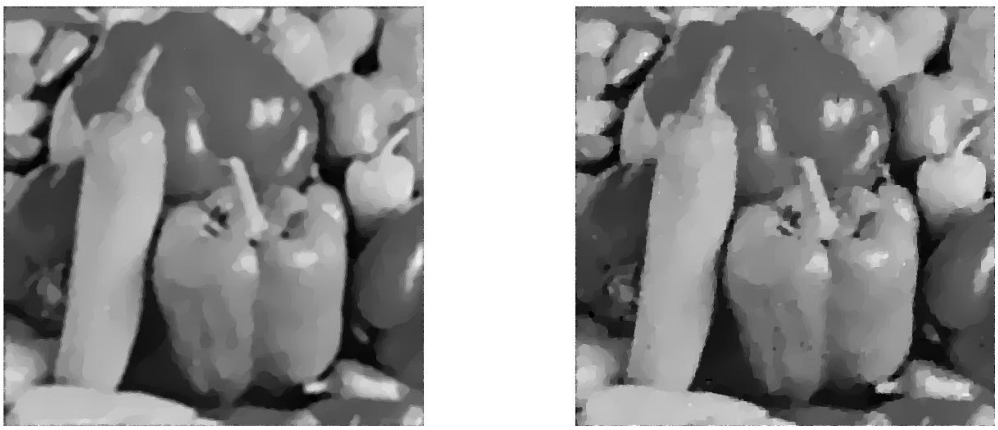
| Noise Type | Model | SSIM | PSNR |
|---|---|---|---|
| Gaussian | Corrupted | 0.15 | 18.79 |
| Gaussian | Abubakar | 0.71 | 25.90 |
| Gaussian | EMM | 0.71 | 26.03 |
| K ($\alpha = 10$) (asymm bell) | Corrupted | 0.16 | 18.85 |
| K ($\alpha = 10$) (asymm bell) | Abubakar | 0.68 | 25.03 |
| K ($\alpha = 10$) (asymm bell) | EMM | 0.69 | 25.70 |
| K ($\alpha = 0.5$)(spiky) | Corrupted | 0.17 | 18.78 |
| K ($\alpha = 0.5$)(spiky) | Abubakar | 0.68 | 25.49 |
| K ($\alpha = 0.5$)(spiky) | EMM | 0.69 | 25.52 |

Table 6.5: SSIM and PSNR index of reconstructed Peppers using Abubakar's multiplicative model and EMM with noise amplitude at 10% level ($\mu = \mu^*_{cor} = 3.37$).



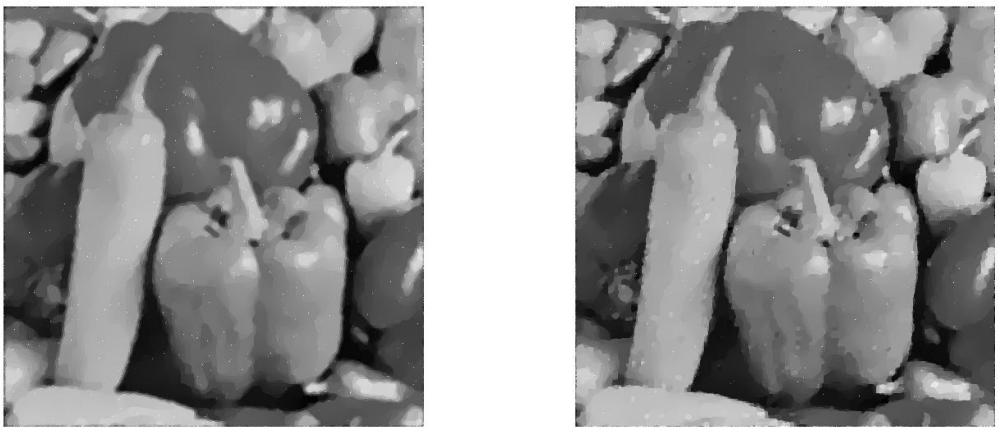
(a) Abubakar's model (b) EMM

Figure 6.12: Reconstructed images at 10% Gaussian noise using Abubakar's model and EMM



(a) Abubakar's model (b) EMM

Figure 6.13: Reconstructed images at 10% K-distributed noise ($\alpha = 10$, asymm bell) using Abubakar's model and EMM

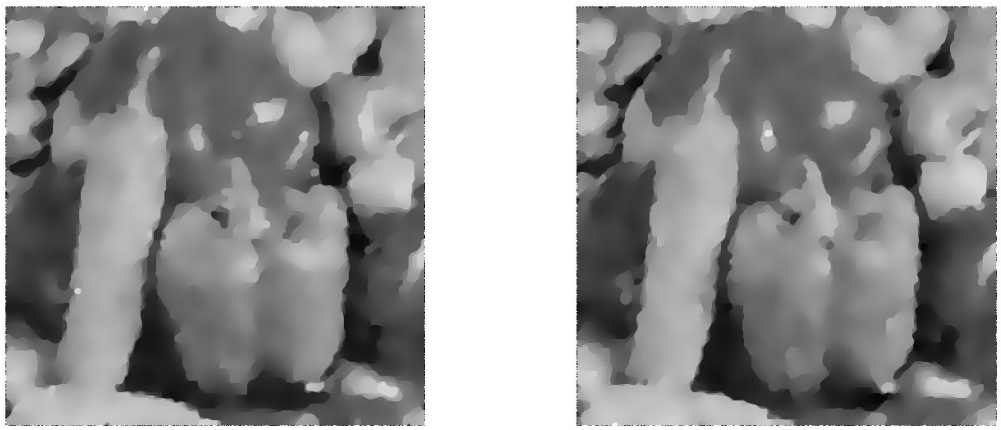


(a) Abubakar's model (b) EMM

Figure 6.14: Reconstructed images at 10% K-distributed noise ($\alpha = 0.5$, spiky) using Abubakar's model and EMM

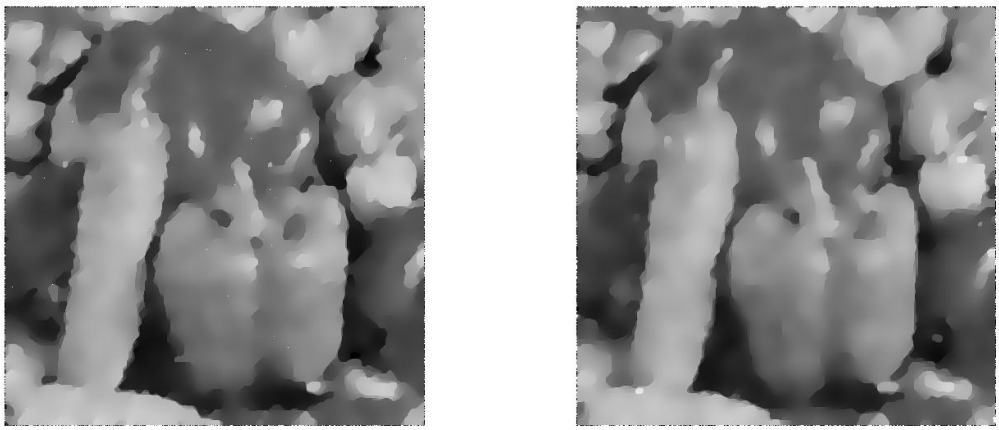
| Noise Type | Model | SSIM | PSNR |
|---|---|---|---|
| Gaussian | Corrupted | 0.01 | 5.98 |
| Gaussian | Abubakar | 0.63 | 21.00 |
| Gaussian | EMM | 0.63 | 21.19 |
| K ($\alpha = 10$) (asymm bell) | Corrupted | 0.01 | 5.77 |
| K ($\alpha = 10$) (asymm bell) | Abubakar | 0.63 | 21.09 |
| K ($\alpha = 10$) (asymm bell) | EMM | 0.63 | 20.77 |
| K ($\alpha = 0.5$)(spiky) | Corrupted | 0.01 | 5.75 |
| K ($\alpha = 0.5$)(spiky) | Abubakar | 0.52 | 17.90 |
| K ($\alpha = 0.5$)(spiky) | EMM | 0.59 | 19.16 |

Table 6.6: SSIM and PSNR index of reconstructed Peppers using Abubakar's multiplicative model and EMM with high noise amplitude at 50% level ($\mu = \mu^* = 8.82$).



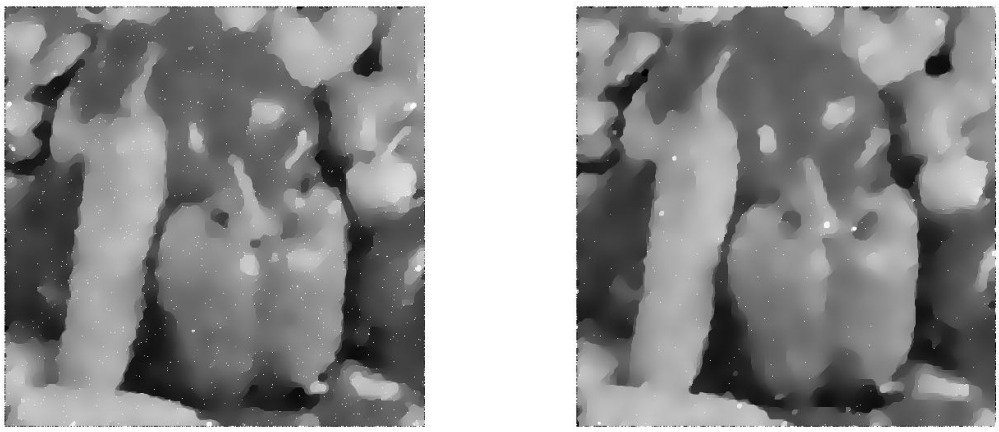
(a) Abubakar's model (b) EMM

Figure 6.15: Reconstructed images at 50% Gaussian noise using Abubakar's model and EMM



(a) Abubakar's model (b) EMM

Figure 6.16: Reconstructed images at 50% K-distributed noise ($\alpha = 10$, asymm bell) using Abubakar's model and EMM



(a) Abubakar's model (b) EMM

Figure 6.17: Reconstructed images at 50% K-distributed noise ($\alpha = 0.5$, spiky) using Abubakar's model and EMM

# Chapter 7

# Conclusion and Future Work

## 7.1 Conclusion

This dissertation is primarily concerned with self-adaptive regularisation methods for inverse problems, where the regularisation strength is automatically adjusted in the model. It starts by giving a general review of the field of inverse problems and common regularisation strategies, then introduces the recent (self-adaptive) multiplicative model by Abubakar *et al* [AvdBHB04]. This model has promising numerical results, especially for piecewise constant image deblurring problems with in the presence of noise with high amplitude and various distributions. However, there is a need for more theoretical results, especially regarding its convergence properties. Also, by reproducing the model and implementing it on various applications, I realised that this method can be usefully improved in a number of ways. Firstly, its performance may be enhanced by inserting an extra tuning parameter $\mu$ in one of the original recurrence relations. An explicit formula for determining the most appropriate value of this tuning parameter is derived in Chapter 4. Applying this formula to determine $\mu$ is significantly less expensive than the tuning of the regularisation parameter in its additive counterparts: we only require a little prior information of the image, which may be obtained by applying a Sobel filter (or equivalent) for edge detection. Equipped with this choice rule for the additional tuning parameter $\mu$, Abubakar's

multiplicative regularisation model is now able to achieve much improved image reconstruction, as well as to avoid the large computational resources that are normally needed in other models for tuning a regularisation parameter. The model is capable of handling large noise level (up to 50%) with various distributions for digital images of all nature.

Secondly, inspired by another self-adaptive model by Charbonnier *et al* [CBFAB97] which is an additive model with regularisation strength spatially controlled by a sequence of weight functions $b_n$ automatically adapted through the algorithm, I have re-formulated the multiplicative method using an alternating minimisation perspective, and successfully constructed new multiplicative models MMR, MSSP and EMM. These models not only remedy the limitation of Abubakar's original multiplicative model [AvdBHB04] in terms of lacking theoretical results, but also inherit its merit of tolerance of high noise level and robustness against different noise distribution. In all of these new multiplicative models, the objective cost functional $C_n$ is monotonically decreasing, and the image intensity $u_n$ exhibits semi-convergence. In addition, MMR and MSSP are able to handle different type of potential functions in the objective cost functional, which can be useful in order to adapt the method to preserving different types of image features, while EMM leads to numerical performance exceeding all the other multiplicative models, both in terms of quantitative and qualitative results.

The following table provides an overview of the different multiplicative models presented in this dissertation, and their corresponding properties:

| Model | $\mu$ | $\delta$ | Algorithm | Convergence |
|---|---|---|---|---|
| Abubakar | Y | self-adapt | 3-step | unknown |
| MMR(HL,GM,HS) | N | const | 2-step | $C_n$ convg, $\|(b^2_{n+1})_{i,j} - (b^2_n)_{i,j}\| \to 0$ [1], $u_n$ semi-convg |
| MSSP(HL) | N | self-adapt | 3-step | $C_n$ convg, $\|(b^2_{n+1})_{i,j} - (b^2_n)_{i,j}\| \to 0$[1], $\|\delta^2_{n+1} - \delta^2_n\| \to 0$[1],$u_n$ semi-convg[1] |
| MSSP(GM,HS) | N | self-adapt | 3-step | $C_n$ convg, $\|\delta^2_{n+1}(b^2_{n+1})_{i,j} - \delta^2_n(b^2_n)_{i,j}\| \to 0$[1], $u_n$ semi-convg |
| EMM | Y | self-adapt | 3-step | $C_n$ convg[2], $\|(b^2_{n+1})_{i,j} - (b^2_n)_{i,j}\| \to 0$[1][2], $u_n$ semi-convg |

Table 7.1: Overview of different multiplicative models

[1] Requires Assumption 5.4.1 which states

$$\sum_{(i,j)} [(\mathcal{K}u_n^{M_n})_{i,j} - f^{\epsilon}_{i,j}]^2 \geq \kappa, \ as \ n \to \infty.$$

[2] Requires choosing the integration constant $M$ such that the objective cost functional $C_n$ does not reach negative values.

In Chapter 6 and 7, I have also discussed the additive counterparts of MMR and MSSP. These models serve as a "bridge", building a connection between Charbonnier's additive model [CBFAB97] and Abubakar's multiplicative model [AvdBHB04], and help with the understanding of all multiplicative models presented. We note that, when comparing MMR and MSSP with their additive counterparts, a conjugate gradient method is applied in the minimisation of image intensity $u$ for these additive models, in order to compare 'like for like" as much as possible. This is different from Charbonnier *et al* [CBFAB97], where a Gauss-Newton method is applied. In addition, Charbonnier model allows the weight functions $b_x$ and $b_y$ to be different with respect to finite differences $D_x u$ and $D_y u$, while in my implementation I set $b_x = b_y = b$ in order to fit the form of the multiplicative models. Both of these changes might weaken the performance of the original Charbonnier model [CBFAB97].

The following table provides another overview of different additive models presented in this dissertation, and their corresponding properties:

| Model | $\delta$ | Algorithm | Convergence |
|---|---|---|---|
| Charbonnier | $\delta \equiv 1$ | 2-step | $C_n$, $b_n$,$u_n$ convg[3] |
| MMR counterpart (HL,GM,HS) | const | 2-step | $C_n$, $b_n$,$u_n$ convg[3] |
| MSSP counterpart (HL) | self-adapt | 3-step | $C_n$ convg, $\lvert(b^2_{n+1})_{i,j} - (b^2_n)_{i,j}\rvert \to 0$, $\lvert\delta^2_{n+1} - \delta^2_n\rvert \to 0$,$u_n$ semi-convg |
| MSSP counterpart (GM,HS) | self-adapt | 3-step | $C_n$ convg, $\lvert\delta^2_{n+1}(b^2_{n+1})_{i,j} - \delta^2_n(b^2_n)_{i,j}\rvert \to 0$ $u_n$ semi-convg |

Table 7.2: Overview of different additive models

[3] In additive models, in the cases where image intensity $u_n$ converges, it is possible to show that this limit is indeed the global minimiser of objective $\bar{C}(u) = \arg\min_b C(u,b)$ or $\tilde{C}(u) = \arg\min_{\delta,b} C(u,b,\delta)$, with the extra assumption that matrix $\mathcal{K}$ has full rank. Detailed proof may be found in Charbonnier *et al* [CBFAB97] and Appendix C (v) of this dissertation. The additive counterpart of MMR is essentially Charbonnier model with a fixed scaling parameter hence the convergence property follows.

The following comments may be added regarding Table 7.1 and Table 7.2:

- At low levels of noise, the merit of robustness towards different noise distributions is shared by all the multiplicative and additive models presented in the tables above. We conclude that this is due to the formulation of potential functions $\varphi_{pot}$.
- None of the additive models has tolerance to high level of noise. Among the multiplicative models, Abubakar's model [AvdBHB04] and EMM perform remarkably well with levels of noise as high as 50%. We conclude that this is due to the augmented recurrence relation which requires the insertion of parameter $\mu$ and its straightforward parameter selection rule as discussed in Chapter 4 of this dissertation .

## 7.2 Future Work

In this section I discuss possible future work regarding these multiplicative regularisation models. This may be directed to the following aspects:

### 7.2.1 Termination Criterion

As discussed in Chapter 5-6, all multiplicative models MMR, MSSP and EMM exhibit semi-convergence of the image intensity $u_n$, and are equipped with an alternating minimisation algorithm in two and three steps (Algorithm 5.2.1). This algorithms employs an arbitrary choice of $M_{outer}$ and $M_{inner}$ to cut off the iteration in both outer and inner loops. The purpose of this is to facilitate the semi-convergence of the image intensity $u_n$: the regularisation model in this case bears the nature of an iterative method with iteration index playing the role of regularisation parameter. Currently $M_{inner}$ and $M_{outer}$ are set to be to be approximately 10-20 and 200-300 respectively by trial and error. This leads to satisfactory numerical results for the implementation presented in this dissertation, nevertheless a more sophisticated choice rule for $M_{inner}$ and $M_{outer}$ is called for in the future, and it is worth exploring ways of establishing better stopping criteria, with better rigorous justification.

### 7.2.2 Physical Interpretation

The multiplicative type of regularisation models are novel methods and so far remain "ad hoc" without clear physical interpretation. In order to find some physical meaning for them, one possible approach would be to relate them to the L-curve method and the additive models as introduced in section 3.1. The objective cost functional of the multiplicative models has a form similar to the corner classification of the L-curve method in (2.2.25) $\psi(\alpha) := ||y^\epsilon - Ax_\alpha^\epsilon||||x_\alpha^\epsilon||$, and it will be a promising perspective to join the theory of both fields in the future.

### 7.2.3 New Recurrence Relation

In Section 6.1 we have seen that it is not possible to extend the EMM model to other types of potential functions by keeping the $\delta$-recurrence relation (6.1.5). This recurrence relation leads to the EMM model in the HL type that (a) outperforms Abubakar's model numerically and (b) inherits the same choice rule for parameter $\mu$ as discussed in Chapter 4. It is worth exploring further on the possible extensions of EMM model to other types of potential functions using new $\delta$-recurrence relations.

### 7.2.4 Applications

Although not included in this dissertation, in my research I also implemented the multiplicative regularisation as formulated by Abubakar *et al* [AvdBHB04] in the application of elastography, to stabilise the ill-posed differential operator $\nabla$ in the wave equation. Unfortunately the numerical results in this application do not appear to be as satisfactory as they are in the deblurring problem with the presence of noise. However, this does not imply the failure of multiplicative regularisation in other applications: as a hybrid inverse problem, elastography is a complicated application, and required in the first instance some drastic simplifications in order to apply multiplicative regularisation. So other factors may have been the main cause preventing the multiplicative model from reaching its full potential in this case. Fewer simplifications may have led to better results. As a novel approach and as a new method successfully used in a number of applications, the implementation of multiplicative model in practice is worth future exploration.

# Appendices

# Appendix A

# Proof of Theorem 3.3.2

*Proof.* The proof presented here may also be found in Charbonnier *et al* [CBFAB97] and Geman and Reynolds [GR92]. To start with, define

$$\theta(t) = \varphi(\sqrt{t}), \tag{A.0.1}$$

and obtain its derivative by chain rule:

$$\theta'(t) = \frac{\varphi'(\sqrt{t})}{2\sqrt{t}}. \tag{A.0.2}$$

Compared with condition (e) of $\varphi(t)$ function, this means that $\theta' : [0, +\infty) \rightarrow (0, M]$ is continuous and strictly decreasing, hence $\theta''(t) < 0$, $\forall t > 0$. In other words, $\theta(t)$ is strictly concave. This leads to the following decomposition:

$$\theta(u) \leq \theta(v) + (u - v)\theta'(v), \ \forall \ u, v \ \in [0, +\infty), \tag{A.0.3}$$

with equality taken if and only if $u = v$. This means

$$\theta(u) = \inf_{v} \{\theta'(v)u + \theta(v) - v\theta'(v)\}. \tag{A.0.4}$$

To interpret this result further, define a change of variable

$$\omega = \theta'(v)$$

which is equivalent to

$$v = (\theta')^{-1}(\omega), \tag{A.0.5}$$

Recall that $\theta'(t)$ is a strictly decreasing function, hence it is one to one and guarantees a strictly increasing inverse $(\theta')^{-1}(\omega) : (0, M] \to [0, +\infty)$, which justifies this change of variable. By defining

$$\psi(\omega) = \theta[(\theta')^{-1}(\omega)] - \omega \cdot (\theta')^{-1}(\omega), \tag{A.0.6}$$

(A.0.4) may be turned into

$$\theta(u) = \inf_{\omega}\{\omega u + \psi(\omega)\}.$$

Setting $u = t^2$, function $\psi(\omega)$ and $\varphi(t)$ are now related as:

$$\varphi(t) = \theta(t^2) = \inf_{\omega}\{\omega t^2 + \psi(\omega)\}, \tag{A.0.7}$$

which shows the existence of (2.5.3). Next we explore the properties of the newly constructed function $\psi(\omega)$ by taking its derivative with respect to $\omega$. Define

$$z = (\theta')^{-1}(\omega),$$

together with equation (A.0.6), we have

$$\begin{aligned} \psi(\omega) &= \theta(z) - \omega z, \\ \theta'(z) &= \omega, \end{aligned} \tag{A.0.8}$$

hence

$$\psi'(\omega) = \theta'(z)\frac{dz}{d\omega} - z - \omega\frac{dz}{d\omega} = -z = -(\theta')^{-1}(\omega).$$

Recall previously that $(\theta')^{-1}(\omega) : (0, M] \to [0, +\infty)$. Hence $\psi'(\omega) < 0,\ \forall\omega \in (0, M]$, and $\psi$ is strictly decreasing. Therefore its limits exist at $0^+$ and $M$, and may be calculated as follows:

$$\alpha = \lim_{\omega\to M} \psi(\omega) = \lim_{v\to 0^+} (\theta(v) - v\theta'(v)),$$

by (A.0.5). From $\theta(t) = \varphi(t^2)$ and condition (a) we may conclude that $\theta(0) = 0$. This results in $\alpha = 0$ together with the boundedness of $\theta'(v)$ on $(0, M]$. The other limit of $\psi$ happens at

$$\beta = \lim_{\omega\to 0^+} \psi(\omega) = \lim_{v\to+\infty} (\theta(v) - v\theta'(v)).$$

By setting $v = t^2$, and substituting eqn (A.0.1) and (A.0.2) into this expression, we have proved (2.5.2) in the theorem statement. To show the rest of the theorem, we consider the convexity of $\psi$, and calculate its second order derivative:

$$\begin{aligned} \psi''(\omega) = \lim_{h\to 0} \frac{\psi'(\omega + h) - \psi'(\omega)}{h} &= \lim_{h\to 0} \frac{-(\theta')^{-1}(\omega + h) + (\theta')^{-1}(\omega)}{(\omega + h) - \omega} \\ &= -\lim_{h\to 0} \frac{(\theta')^{-1}(\omega) - (\theta')^{-1}(\omega + h)}{\theta'[(\theta')^{-1}(\omega)] - \theta'[(\theta')^{-1}(\omega + h)]} \end{aligned}$$

Recall definition $v = (\theta')^{-1}(\omega)$. By introducing another variable $q = (\theta')^{-1}(\omega + h)$,

this expression may be simplified to

$$\psi''(\omega) = \lim_{q \to v} \frac{v - q}{\theta'(v) - \theta'(q)} = -\frac{1}{\theta''(v)} \tag{A.0.9}$$

$\theta'$ being strictly decreasing implies $\theta''(v) < 0, \ \forall v > 0$, which suggests the strictly concavity of $\theta$ as well as the strict convexity of $\psi$. So far Theorem statement (1) has been proved. To show part (2) of the theorem, recall from (A.0.7) and (A.0.3):

$$\inf_{0<\omega\leq M}(\omega t^2 + \psi(\omega)) = \theta(t^2) \leq \theta(v) + (t^2 - v)\theta'(v),$$

with transformation

$$v = (\theta')^{-1}(\omega) \tag{A.0.10}$$

and

$$u = t^2,$$

with equality happens if and only if

$$v = t^2. \tag{A.0.11}$$

The transformation of (A.0.10) and (A.0.11) gives the uniqueness of the minimiser at

$$\omega_t = \theta'(v) = \theta'(t^2) = \frac{\varphi'(t)}{2t}, \tag{A.0.12}$$

which completes the proof.

□

# Appendix B

# Proof of Theorem 3.3.5

*Proof.* Recall that by construction

$$J(u) = \min_b J^*(u, b),$$

hence

$$J(u) \leq J^*(u, b),\ \forall u,\ \forall b.$$

By definition

$$G(b) = \min_u J^*(u, b),$$

by swapping order of the variables

$$\min_u J(u) = \min_{u,b} J^*(u, b) = \min_{b,u} J^*(u, b) = \min_b G(b),$$

By definition of $\hat{b}$

$$\min_u J(u) = \min_b G(b) = G(\hat{b}),$$

hence there exists a $\hat{u}_{\hat{b}}$ which minimises $J(u)$, in particular

$$J(\hat{u}_{\hat{b}}) \leq J^*(\hat{u}_{\hat{b}}, \hat{b}), \tag{B.0.1}$$

By the definition of $\hat{b}$, we may conclude that

$$J^*(\hat{u}_{\hat{b}}, \hat{b}) \leq \min_u J^*(u, b), \ \forall b.$$

Hence

$$J^*(\hat{u}_{\hat{b}}, \hat{b}) \leq \min_b \min_u J^*(u, b) = \min_u \min_b J^*(u, b) = \min_u J(u) \tag{B.0.2}$$

Combine (B.0.1) and (B.0.2) we may conclude that

$$J(\hat{u}_{\hat{b}}) \leq \min_u J(u).$$

Hence

$$J(\hat{u}_{\hat{b}}) = \min_u J(u),$$

*i.e.*

$$\hat{u}_{\hat{b}} = \arg\min_u J(u)$$

which completes the proof. □

# Appendix C

# Proof of Theorem 3.3.6

*Proof.* **(i) Convergence of objective functional** $J^*(u_n, b_{n+1})$
Firstly we show that the sequence of objective functional $J^*(u_n, b_{n+1})$ is a monotonic descending chain, *i.e.*

$$J^*(u_n, b_{n+1}) \leq J^*(u_{n-1}, b_n), \ \forall u, b. \tag{C.0.1}$$

To start with, recall that $b_{n+1}$ minimises $J^*(u_n, b)$ in Algorithm 2.5.3, hence

$$J^*(u_n, b_{n+1}) \leq J^*(u_n, b), \ \forall b, n$$

Substitute $b = b_n$ into the expression above:

$$J^*(u_n, b_{n+1}) \leq J^*(u_n, b_n), \ \forall b, n. \tag{C.0.2}$$

Similarly, in Algorithm 2.5.3 $u_n$ minimises $J^*(u, b_n)$ at $n$th step, then

$$J^*(u_n, b_n) \leq J^*(u_{n-1}, b_n), \ \forall n. \tag{C.0.3}$$

Hence (C.0.2) and (C.0.3) together shows

$$J^*(u_n, b_{n+1}) \leq J^*(u_{n-1}, b_n), \ \forall n. \tag{C.0.4}$$

It is obvious that as an objective cost functional, $J^*(u_n, b_{n+1}) \geq 0, \ \forall n$, as both the fidelity term and the potential function $\varphi$ in the regularisation are non-negative. As a monotonically decreasing sequence bounded below, sequence $J^*(u_n, b_{n+1})$ is convergent.

**(ii) Convergence of auxiliary variable $b_n$**

In order to show

$$[b_{n+1} - b_n]_{n\to\infty} \to 0, \ \forall n,$$

which translates pixel-wisely to

$$|(b_{n+1})_{i,j} - (b_{n+1})_{i,j}| \to 0, \ \forall i, j. \tag{C.0.5}$$

Let us calculate

$$\begin{aligned} & J^*(u_n, b_n) - J^*(u_n, b_{n+1}) \\ = \ & \lambda \sum_{(i,j)} \{(b_x^n)_{i,j}(D_x u_n)_{i,j}^2 + (b_y^n)_{i,j}(D_y u_n)_{i,j}^2 + \psi[(b_x^n)_{i,j}] + \psi[(b_y^n)_{i,j}]\} \\ & -\lambda \sum_{(i,j)} \{(b_x^{n+1})_{i,j}(D_x u_n)_{i,j}^2 + (b_y^{n+1})_{i,j}(D_y u_n)_{i,j}^2 + \psi[(b_x^{n+1})_{i,j}] + \psi[(b_y^{n+1})_{i,j}]\} \end{aligned} \tag{C.0.6}$$

Define functions

$$\begin{aligned} (p_x^n)_{i,j}(\omega) &= \omega(D_x u_n)_{i,j}^2 + \psi(\omega), \\ (p_y^n)_{i,j}(\omega) &= \omega(D_y u_n)_{i,j}^2 + \psi(\omega), \end{aligned}$$

Then by Taylor expansion at $\omega = (b_x^{n+1})_{i,j}$ and $\omega = (b_y^{n+1})_{i,j}$ in $(p_x^n)_{i,j}(\omega)$ and

$(p_y^n)_{i,j}(\omega)$ respectively, (C.0.6) becomes

$$\begin{aligned} & J^*(u_n, b_n) - J^*(u_n, b_{n+1}) \\ = \; & \lambda \sum_{(i,j)} \{[(b_x^n)_{i,j} - (b_x^{n+1})_{i,j}](p_x^n)'_{i,j}((b_x^{n+1})_{i,j}) + [(b_y^n)_{i,j} - (b_y^{n+1})_{i,j}](p_y^n)'_{i,j}((b_y^{n+1})_{i,j})\} \\ & + \frac{\lambda}{2} \sum_{(i,j)} \{[(b_x^n)_{i,j} - (b_x^{n+1})_{i,j}](p_x^n)''_{i,j}((c_x^n)_{i,j}) + [(b_y^n)_{i,j} - (b_y^{n+1})_{i,j}](p_y^n)''_{i,j}((c_y^n)_{i,j})\}, \quad \text{(C.0.7)} \end{aligned}$$

where $(c_x^n)_{i,j} \in \big((b_x^n)_{i,j}, (b_x^{n+1})_{i,j}\big)$, and $(c_y^n)_{i,j} \in \big((b_y^n)_{i,j}, (b_y^{n+1})_{i,j}\big)$. Recall from Algorithm 2.5.3 that $b^{n+1}$ minimises $J^*(u_n, b)$, hence

$$\begin{aligned} (p_x^n)'_{i,j}((b_x^{n+1})_{i,j}) &= (D_x u_n)^2_{i,j} + \psi'((b_x^{n+1})_{i,j}) = \frac{\partial}{\partial b} J^*(u_n, b)\big|_{b=(b_x^{n+1})_{i,j}} = 0, \\ (p_y^n)'_{i,j}((b_y^{n+1})_{i,j}) &= (D_y u_n)^2_{i,j} + \psi'((b_y^{n+1})_{i,j}) = \frac{\partial}{\partial b} J^*(u_n, b)\big|_{b=(b_y^{n+1})_{i,j}} = 0. \quad \text{(C.0.8)} \end{aligned}$$

Hence (C.0.7) now becomes

$$\begin{aligned} & J^*(u_n, b_n) - J^*(u_n, b_{n+1}) \\ = \; & \frac{\lambda}{2} \sum_{(i,j)} \{[(b_x^n)_{i,j} - (b_x^{n+1})_{i,j}](p_x^n)''_{i,j}((c_x^n)_{i,j}) + [(b_y^n)_{i,j} - (b_y^{n+1})_{i,j}](p_y^n)''_{i,j}((c_y^n)_{i,j})\}. \quad \text{(C.0.9)} \end{aligned}$$

Our next aim is to come up with an upper bound for $(p_x^n)''_{i,j}(\omega)$ and $(p_y^n)''_{i,j}(\omega)$, $\forall \omega$. Note that

$$(p_x^n)''_{i,j}(\omega) = (p_y^n)''_{i,j}(\omega) = \psi''(\omega), \quad \text{(C.0.10)}$$

Recall from (A.0.9) in Theorem 2.5.1 that

$$\psi''(\omega) = -\frac{1}{\theta''(t^2)}, \quad \text{(C.0.11)}$$

with variable $\omega$ and $t$ related via

$$\omega = \theta'(t^2) = \frac{\varphi'(t)}{2t} \tag{C.0.12}$$

as defined previously. This means that in order to bound $p''_{i,j}(\omega)$as well as $\psi''(\omega)$, we can simply work in the space of $t$, and aim to bound $-\theta''(t^2)$ above for $t \in (0, +\infty)$. This approach enables a possibility to make use of the potential function $\varphi(t)$, which allows us to benefit from its extensive assumptions (a)-(i) in Theorem 2.5.1.

To proceed with the calculation for $\theta''(t^2)$, we differentiate (C.0.12) with respect to $t$, and apply the change of variable $v = t^2$ defined in previous sections:

$$\theta'(v) = \frac{\varphi'(\sqrt{v})}{2\sqrt{v}},$$

hence

$$\theta''(v) = \frac{\varphi''(\sqrt{v})}{4\sqrt{v}} - \frac{\varphi'(\sqrt{v})}{4v\sqrt{v}},$$

back to variable $t$:

$$-\theta''(t^2) = \frac{1}{2t^2}\left(\frac{\varphi'(t)}{2t} - \frac{\varphi''(t)}{2}\right). \tag{C.0.13}$$

In order to bound $-\theta''(t^2)$ above for all $t \in (0, +\infty)$, we need to explore the property of the function it $t$ on the right hand side of (C.0.13). Note that

$$\left(\frac{\varphi'(t)}{2t}\right)' = -\frac{1}{2t}\left(\frac{\varphi'(t)}{t} - \varphi''(t)\right), \tag{C.0.14}$$

Assumption (e)-(f), tells us that $\frac{\varphi'(t)}{2t}$ is continuously and strictly decreasing, with limit tends to 0 as $t \to +\infty$. This means that its derivative stays negative for all

$t \in (0, +\infty)$, and necessarily vanishes as $t \to +\infty$. This leads to

$$\lim_{t\to+\infty} -\frac{1}{2t}\left(\frac{\varphi'(t)}{t} - \varphi''(t)\right) = \lim_{t\to+\infty}\left(\frac{\varphi'(t)}{2t}\right)' = 0, \tag{C.0.15}$$

hence

$$\lim_{t\to+\infty} -\theta''(t^2) = \lim_{t\to+\infty} \frac{1}{2t^2}\left(\frac{\varphi'(t)}{2t} - \frac{\varphi''(t)}{2}\right) = 0. \tag{C.0.16}$$

On the other hand, we investigate how $-\theta''(t^2)$ as $t \to 0^+$. We stick with expression (C.0.13), and apply Taylor expansion for both $\frac{\varphi'(t)}{2t}$ and $\frac{\varphi''(t)}{2}$ at $t = 0$:

$$\begin{aligned}\varphi'(t) &= \varphi'(0) + \varphi''(0)t + \frac{1}{2}\varphi'''(0)t^2 + \frac{1}{6}\varphi^{(4)}(0)t^3 + \epsilon(t^4)\\ \varphi''(t) &= \varphi''(0) + \varphi'''(0)t + \frac{1}{2}\varphi^{(4)}(0)t^2 + \epsilon(t^3)\end{aligned}$$

We may tell from either assumptions (b)(c) for $\varphi$ or assumption (g) for $\frac{\varphi'(t)}{2t}$ that it is necessary for $\varphi'(0) = 0$. Also assmption (h) specifies $\varphi'''(0) = 0$. Hence in the neighbourhood of $t = 0$,

$$\frac{1}{2t^2}\left(\frac{\varphi'(t)}{2t} - \frac{\varphi''(t)}{2}\right) = -\frac{1}{12}\varphi^{(4)}(0) + \epsilon(t), \tag{C.0.17}$$

with existence of $\varphi^{(4)}(0)$ guaranteed in assumption (i). This means that function $\theta''(t^2)$ is finite when $t \to 0^+$. Thus $-\theta''(t^2)$ may be bounded above for all $t \in [0, +\infty)$. Recall (C.0.11), this is equivalently

$$\begin{aligned}|(p_x^n)''_{i,j}((c_x^n)_{i,j})| &= |\psi''((c_x^n)_{i,j})| \geq \frac{1}{C},\\ |(p_y^n)''_{i,j}((c_y^n)_{i,j})| &= |\psi''((c_x^n)_{i,j})| \geq \frac{1}{C},\end{aligned} \tag{C.0.18}$$

Together with (C.0.9),

$$\begin{aligned} & J^*(u_n, b_n) - J^*(u_n, b_{n+1}) \\ \geq \ & \frac{\lambda}{2C} \sum_{(i,j)} \{[(b_x^n)_{i,j} - (b_x^{n+1})_{i,j}] + [(b_y^n)_{i,j} - (b_y^{n+1})_{i,j}]\}. \end{aligned} \tag{C.0.19}$$

As it has been shown previously that $\{J^*(u_n, b_n)\}_{n\in\mathbb{N}}$ is convergent, and $J^*(u_n, b_n) \geq J^*(u_n, b_{n+1}) \geq J^*(u_{n+1}, b_{n+1})$, hence $J^*(u_n, b_n) - J^*(u_n, b_{n+1}) \to 0,\ n \to \infty$, which forces

$$\begin{aligned} (b_x^n)_{i,j} - (b_x^{n+1})_{i,j} \to 0,\ n \to \infty \\ (b_y^n)_{i,j} - (b_y^{n+1})_{i,j} \to 0,\ n \to \infty \end{aligned}$$

thus completes the proof.

**(iii) Convergence of image intensity $u_n$ to the global minimiser $u_{\hat{b}}$ (when $\mathcal{K}$ has full rank):**

In this part of the proof, we try to show that with the extra assumptions of $\varphi$ being convex and $\mathcal{K}$ having full rank, Algorithm 2.5.3 will indeed converge to the global minimiser $u_{\hat{b}}$ of the objective cost functional $J(u) = \inf_b J^*(u, b)$. Recall for each fixed $b^n$, $J^*(u, b^n)$ is convex in $u$, and minimal is attained at $u = u_n$. This leads to the first order optimality condition by fixing $b^n$ in $J^*(u, b^n)$ and differentiating it with respect to image intensity $u$:

$$(\mathbf{K}^T\mathbf{K} + \lambda\mathbf{D}^T\mathbf{B}^n\mathbf{D})\mathbf{u}_n - \mathbf{K}^T\mathbf{f}^\epsilon = 0, \tag{C.0.20}$$

In this expression, we re-shape the 2D image intensity $u_n$ and $f^\epsilon$from $N \times N$ matrices to 1D arrays with dimension $1 \times N^2$. Note that $\mathbf{K}$, $\mathbf{D}$ and $\mathbf{B}$ are all $N^2 \times N^2$ matrices. They represent the convolution operator $\mathcal{K}$, the finite difference operator $D$ and the weight function $b_x$, $b_y$ correspondingly in this setting.

Similarly, as we have shown that $u_{\hat{b}}$ is the unique global minimiser of the objective cost functional $J(u) = \inf_b J^*(u, b)$ (due to the convexity of $J(u)$ in $u$), and it satisfies

the first order condition necessarily:

$$(\mathbf{K}^T\mathbf{K} + \lambda\mathbf{D}^T\hat{\mathbf{B}}\mathbf{D})\mathbf{u}_{\hat{b}} - \mathbf{K}^T\mathbf{f}^{\epsilon} = 0, \tag{C.0.21}$$

where $\hat{\mathbf{B}}$ is the $N^2 \times N^2$ matrix representation regarding auxiliary variables $\hat{b}_x$, $\hat{b}_y$ correspondingly.

Subtracting (C.0.20) and (C.0.21) and making scalar product with $u_n - u_{\hat{b}}$, we obtain

$$||\mathbf{K}(\mathbf{u}_n - \mathbf{u}_{\hat{b}})||^2 + \lambda\big\langle \mathbf{B}^n\mathbf{D}\mathbf{u}_n - \hat{\mathbf{B}}\mathbf{D}\mathbf{u}_{\hat{b}}, \mathbf{D}(\mathbf{u}_n - \mathbf{u}_{\hat{b}})\big\rangle = 0 \tag{C.0.22}$$

By inserting term $\pm\lambda\big\langle \mathbf{B}^{n+1}\mathbf{D}\mathbf{u}_n, \mathbf{D}(\mathbf{u}_n - \mathbf{u}_{\hat{b}})\big\rangle$, (C.0.22) becomes

$$\begin{aligned} ||\mathbf{K}(\mathbf{u}_n - \mathbf{u}_{\hat{b}})||^2 + \lambda\big\langle \mathbf{B}^{n+1}\mathbf{D}\mathbf{u}_n - \hat{\mathbf{B}}\mathbf{D}\mathbf{u}_{\hat{b}}, \mathbf{D}(\mathbf{u}_n - \mathbf{u}_{\hat{b}})\big\rangle \\ +\lambda\big\langle \mathbf{B}^n\mathbf{D}\mathbf{u}_n - \mathbf{B}^{n+1}\mathbf{D}\mathbf{u}_n, \mathbf{D}(\mathbf{u}_n - \mathbf{u}_{\hat{b}})\big\rangle = 0 \end{aligned} \tag{C.0.23}$$

Recall Remark 2.5.2, for each pixel $(i, j)$,

$$(\hat{b}_x)_{i,j}(D_x u_{\hat{b}})_{i,j} = \frac{1}{2}\varphi'[(D_x u_{\hat{b}})_{i,j}],\ (\hat{b}_y)_{i,j}(D_y u_{\hat{b}})_{i,j} = \frac{1}{2}\varphi'[(D_y u_{\hat{b}})_{i,j}], \forall\ i, j \tag{C.0.24}$$

$$(b_x^{n+1})_{i,j}(D_x u_n)_{i,j} = \frac{1}{2}\varphi'[(D_x u_n)_{i,j}],\ (b_y^{n+1})_{i,j}(D_y u_n)_{i,j} = \frac{1}{2}\varphi'[(D_y u_n)_{i,j}].\forall\ i, j \tag{C.0.25}$$

With the assumption of potential functional $\varphi(t)$ being convex, *i.e.*

$$[\varphi'(u) - \varphi'(v)] \cdot (u - v) \geq 0,$$

we have

$$[\varphi'((D_x u_n)_{i,j}) - \varphi'((D_x u_{\hat{b}})_{i,j})] \cdot ((D_x u_n)_{i,j} - (D_x u_{\hat{b}})_{i,j}) \geq 0, \forall\ i, j \qquad \text{(C.0.26)}$$

$$[\varphi'((D_y u_n)_{i,j}) - \varphi'((D_y u_{\hat{b}})_{i,j})] \cdot ((D_y u_n)_{i,j} - (D_y u_{\hat{b}})_{i,j}) \geq 0. \forall\ i, j \qquad \text{(C.0.27)}$$

Summarize over all the pixels, (C.0.24)-(C.0.27) lead to

$$\left\langle \mathbf{B}^{n+1}\mathbf{D}\mathbf{u}_n - \hat{\mathbf{B}}\mathbf{D}\mathbf{u}_{\hat{b}}, \mathbf{D}(\mathbf{u}_n - \mathbf{u}_{\hat{b}}) \right\rangle > 0, \qquad \text{(C.0.28)}$$

together with (C.0.23), we obtain

$$||\mathbf{K}(\mathbf{u}_n - \mathbf{u}_{\hat{b}})||^2 + \lambda \left\langle \mathbf{B}^n\mathbf{D}\mathbf{u}_n - \mathbf{B}^{n+1}\mathbf{D}\mathbf{u}_n, \mathbf{D}(\mathbf{u}_n - \mathbf{u}_{\hat{b}}) \right\rangle < 0,$$

By Schwartz Inequality, we have

$$\begin{aligned} ||\mathbf{K}(\mathbf{u}_n - \mathbf{u}_{\hat{b}})||^2 &< \lambda \left| \left\langle \mathbf{B}^n\mathbf{D}\mathbf{u}_n - \mathbf{B}^{n+1}\mathbf{D}\mathbf{u}_n, \mathbf{D}(\mathbf{u}_n - \mathbf{u}_{\hat{b}}) \right\rangle \right| \\ &< \lambda ||\mathbf{B}^n - \mathbf{B}^{n+1}|| \cdot ||\mathbf{D}\mathbf{u}_n|| \cdot ||\mathbf{D}(\mathbf{u}_n - \mathbf{u}_{\hat{b}})||, \end{aligned} \qquad \text{(C.0.29)}$$

As $\mathbf{D}\mathbf{u}_n$ finite difference operator in the discrete setting with convergence result shown in (iii), it is possible to conclude that both terms $||\mathbf{D}\mathbf{u}_n||$ and $||\mathbf{D}(\mathbf{u}_n - \mathbf{u}_{\hat{b}})||$ are finite. Hence there exists $E > 0$ such that $||\mathbf{D}\mathbf{u}_n|| \cdot ||\mathbf{D}(\mathbf{u}_n - \mathbf{u}_{\hat{b}})|| \leq E$, *i.e.*

$$||\mathbf{K}(\mathbf{u}_n - \mathbf{u}_{\hat{b}})||^2 < \lambda E ||\mathbf{B}^n - \mathbf{B}^{n+1}||.$$

As we have shown in (ii) that the auxiliary variable $b$ converges, it may be concluded that $||\mathbf{B}^n - \mathbf{B}^{n+1}|| \to 0,\ n \to \infty$. Hence

$$||\mathbf{K}(\mathbf{u}_n - \mathbf{u}_{\hat{b}})||^2 \to 0,\ n \to \infty. \qquad \text{(C.0.30)}$$

Together with the assumption of full rank of $\mathcal{K}$ and $\mathbf{K}$, it may be concluded from (C.0.30) that $u_n \to u_{\hat{b}},\ n \to \infty$.

**(iv) Convergence of finite difference $Du_n$ to the "correct" limit $Du_{\hat{b}}$ (when $\mathcal{K}$ does not have full rank):**
In the cases where $\mathcal{K}$ and $\mathbf{K}$ does not have full rank, (C.0.30) still holds. However, it is no longer possible to conclude $u_n \to u_{\hat{b}},\ n \to \infty$ from this expression. Instead of showing the image intensity $u_n$ converges to $u_{\hat{b}}$, we now aim to show that the finite difference of the image gradient $D_x u_n \to D_x u_{\hat{b}}$ and $D_y u_n \to D_y u_{\hat{b}}$.

Let us tidy up from previous conclusions: (C.0.29) continues to hold when $\mathcal{K}$ and $\mathbf{K}$ does not have full rank, which means $||\mathbf{K}(\mathbf{u}_n - \mathbf{u}_{\hat{b}})||^2 \to 0$ and $\left\langle \mathbf{B}^n\mathbf{D}\mathbf{u}_n - \mathbf{B}^{n+1}\mathbf{D}\mathbf{u}_n, \mathbf{D}(\mathbf{u}_n - \mathbf{u}_{\hat{b}})\right\rangle \to 0$ when $n \to \infty$. Together with (C.0.23), hence

$$\left\langle \mathbf{B}^{n+1}\mathbf{D}\mathbf{u}_n - \hat{\mathbf{B}}\mathbf{D}\mathbf{u}_{\hat{b}}, \mathbf{D}(\mathbf{u}_n - \mathbf{u}_{\hat{b}})\right\rangle \to 0. \tag{C.0.31}$$

This may also be expressed pixel-wisely as

$$\begin{aligned}\sum_{(i,j)}\{&[\varphi'((D_x u_n)_{i,j}) - \varphi'((D_x u_{\hat{b}})_{i,j})] \cdot ((D_x u_n)_{i,j} - (D_x u_{\hat{b}})_{i,j}) \\ &+[\varphi'((D_y u_n)_{i,j}) - \varphi'((D_y u_{\hat{b}})_{i,j})] \cdot ((D_y u_n)_{i,j} - (D_y u_{\hat{b}})_{i,j})\} \to 0.\end{aligned} \tag{C.0.32}$$

Apply a theorem on convex analysis, which may be found in standard text books such as [Roc97]:

**Theorem C.0.1.** *Let $T : \mathcal{R}^n \to \mathcal{R}$ be a strictly convex and $C^1$ function, then $\forall M > 0$, there exists a function $\rho_M : [0, 2M] \to \mathcal{R}^+$ which is continuous, strictly increasing with $\rho_M(0) = 0$ and*

$$\left\langle T'(v) - T'(u), v - u\right\rangle \geq \rho_M(||v - u||),\ \forall\ u, v \text{ such that } ||u|| \leq M, ||v|| \leq M.$$

Back to our proof: we have shown previously that the finite differences $D_x u_n$,$D_y u_n$ and $D_x u_{\hat{b}},\ D_y u_{\hat{b}}$ are finite and hence bounded above. Apply Theorem C.0.1 to

(C.0.32), there exists a function $\rho_M$ such that

$$\begin{aligned}\sum_{(i,j)}\{&[\varphi'((D_x u_n)_{i,j}) - \varphi'((D_x u_{\hat{b}})_{i,j})] \cdot ((D_x u_n)_{i,j} - (D_x u_{\hat{b}})_{i,j}) \\ &+[\varphi'((D_y u_n)_{i,j}) - \varphi'((D_y u_{\hat{b}})_{i,j})] \cdot ((D_y u_n)_{i,j} - (D_y u_{\hat{b}})_{i,j})\} \\ &\geq \rho_M(||D_x u_n - D_x u_{\hat{b}}||) + \rho_M(||D_y u_n - D_y u_{\hat{b}}||) \end{aligned} \tag{C.0.33}$$

Both (C.0.32) and (C.0.33) implies

$$\begin{aligned}\lim_{n\to+\infty} \rho_M(||D_x u_n - D_x u_{\hat{b}}||) \to 0, \\ \lim_{n\to+\infty} \rho_M(||D_y u_n - D_y u_{\hat{b}}||) \to 0.\end{aligned}$$

As $\rho_M$ is continuous and strictly increasing with $\rho_M(0) = 0$, it is necessarily to have $||D_x u_n - D_x u_{\hat{b}}|| \to 0$ and $||D_y u_n - D_y u_{\hat{b}}|| \to 0$, which completes the theorem.

□

# Appendix D

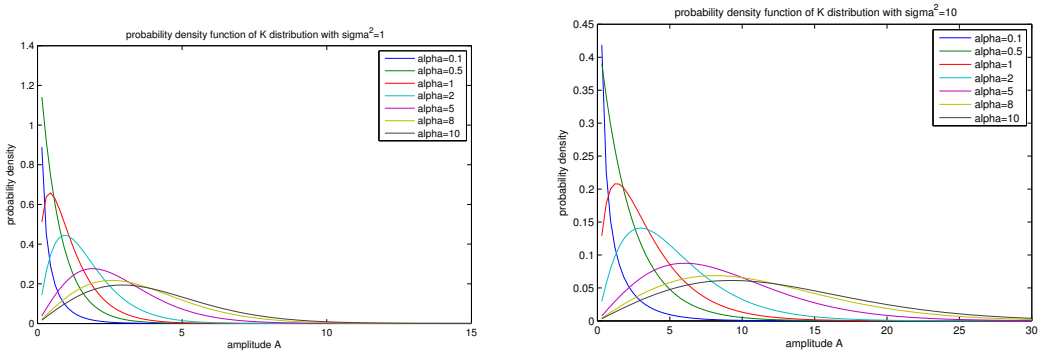
# K-distributed Noise

In this appendix I will discuss the K-distribution, which is used in the numerical results presented in the thesis as an example of asymmetric distributed noise. The K-distribution is a two-parameter distribution with different shapes as the parameters vary, and often used to model ultrasound echo envelope signals [DG94]. According to Jakeman and Pusey [JP76], ( and see also [CD11]), the K-distribution is defined by the probability density function

$$P_K(A|\sigma^2,\alpha) = \frac{4A^{\alpha-1+n/2}}{(2\sigma^2)^{\alpha+n/2}\Gamma(\alpha)\Gamma(n/2)} K_{\alpha-n/2}(\sqrt{\frac{2}{\sigma^2}}A)$$

where
$A$: amplitude of the signal;
$\Gamma$: Euler gamma function;
$K_p$: modified Bessel function of the second kind of order $p$;
$n$: dimension of the random walk taken. (WLOG taken to be 2.)
$\alpha > 0$ and $\sigma^2 > 0$ are respectively a shape parameter and a scaling parameter, and characterise each K-distribution implemented, denoted by $K(\alpha, \sigma^2)$, as illustrated in Figure (D.1).

It may be concluded from Fig D.1 above that the shape of K-distribution is purely




(a) $\sigma^2$=1 (b) $\sigma^2$=10

Figure D.1: Probability density function of K-distribution with different combination of parameters

determined by the shape parameter $\alpha$. Varying the scaling parameter $\sigma$ has no effect on the shape of the distribution at all. The choice of $\sigma$ only determines how disperse the distribution is. Based on this observation, it therefore makes sense to fix $\alpha$ to specify the type of the noise distribution, and adjust the scaling parameter $\sigma$ to give the noise level as required in the model. For my numerical implementation in the next section, I choose to test the K-distribution with shape parameter $\alpha = 0.5$ (spiky on one end) and $\alpha = 10$ (asymmetric bell). These two $\alpha$ values give totally different behaviour of noise, and it is therefore worth using them for the exploration of multiplicative regularisation. As previously noted in Section 3.3, we use the standard deviation of the noise distribution as the metric for the noise level. In the spirit of this, the scaling parameter $\sigma$ is chosen in such a way that the standard deviation of the distribution meets the required noise level. We will also need to shift the distribution function so that it has zero mean. This is important in image processing, and needs to be guaranteed in order to reconstruct the original image back.

Since there are no ready generators for K-distributed random noise, we need to generate K-distributed random variables using the strategy below:

Suppose that $F_K(A|\sigma^2, \alpha)$ is the distribution function of K-distributed random vari-

able A, with shape parameter $\alpha$ and scale parameter $\sigma^2$, i.e.

$$F_K(A|\alpha,\sigma^2) = \int_0^A P_K(A'|\alpha,\sigma^2)dA' \tag{D.0.1}$$

Now we generate a uniformly distributed random variable $S$, and define

$$\begin{aligned} S &\sim unif[0,1], \\ A &= F_K^{-1}(S|\alpha,\sigma^2) \end{aligned}$$

**Lemma D.0.1.**

$$A \sim K(\alpha,\sigma^2),$$

*i.e. $S$ is a uniform random generator for K-distributed random variable $A$.*

*Proof.*

$$\begin{aligned} & P(S \le s) = s \in [0,1] \\ \iff & P(F_K(A|\sigma^2,\alpha) \le s) = s \\ \iff & P(A \le F^{-1}(s)) = s \\ \iff & P(A \le t) = F(t), \end{aligned} \tag{D.0.2}$$

(D.0.3)

with a change of variables $t = F^{-1}(s), s = F(t)$. □

In the application, we wish to implement K-distributed random noise to our test image at each pixel. We will require the random variable to have the following properties:

(i) Zero mean. This is a basic requirement for any distribution of random noise. Oth-

erwise this is equivalent to shifting the value of the original image by the mean of the distribution. In practice this is realised by shifting the corresponding K-distributed random variable by its expectation, which may be obtained numerically.

(ii) Given a K-distributed random variable $A \sim K(\alpha, \sigma^2)$ for a fixed shape parameter $\alpha$, how to choose its scale parameter $\sigma$ such that its standard deviation matches some given value $0 < \epsilon < 1$ which plays the role of noise amplitude. This is equivalent to saying that, instead of characterising each K-distribution by shape parameter $\alpha$ and scale parameter $\sigma^2$, we now use $\alpha$ and $\epsilon$ instead.

To obtain K-distributed random variables satisfying properties (i) and (ii) with shape parameter $\alpha = 0.5$ and 10 and some given standard deviation (noise level) $\epsilon$, the following is implemented in my dissertation:

**Algorithm D.0.2.** ***K-distributed random generator (discrete case)***
*Step 0: Input: shape parameter $\alpha$ and standard deviation $\epsilon$. Calculate their corresponding scale parameter $\sigma^2$, mean value $\mu_{\alpha,\sigma^2}$, and the inversion of cumulated distribution function $F_K^{-1}(S|\alpha, \sigma^2)$ numerically.*

*Step 1: Choose a sufficiently large integer $N$ (in my implementation $N = 10000$), calculated values $F_K^{-1}(\frac{s}{N}|\alpha, \sigma^2),\ \forall s \in \{1, 2, ..., N\}$. Store these values in some database $D$.*

*Step 2: Generate discrete uniformly distributed random variable $S$ such that*

$$Prob(S = s) = \frac{1}{N},\ \forall s \in \{1, 2, ..., N\}.$$

*When $S = s$, take its corresponding value $a_s^{\alpha,\sigma^2} = F_K^{-1}(\frac{s}{N}|\alpha, \sigma^2)$ from database $D$.*

*Step 3: implement value $a_s^{\alpha,\sigma^2} - \mu_{\alpha,\sigma^2}$ for the K-distributed random variable $A$.*

In the case where $N \to \infty$, $S \sim unif[0,1]$ as a consequence, hence $A \sim K(\alpha, \sigma^2) - \mu_{\alpha,\sigma^2}$.

# Bibliography


[ABRS10] H. Attouch, J. Bolte, P. Redont, and A. Soubeyran. Proximal Alternating Minimization and Projection Methods for Nonconvex Problems: An Approach Based on the Kurdyka-Łojasiewicz Inequaltiy. *Mathematics of Operational Research*, 35(2):438–457, May 2010.

[AFP00] L. Ambrosio, N. Fusco, and D. Pallara. *Functions of Bounded Variation and Free Discontinuity Problems*. Clarendon Press, Oxford, 2000.

[AK06] G. Aubert and P. Kornprobst. *Mathematical Problems in Image Processing, Partial Differential Equations and the Calculus of Variations*. Springer, New York, 2006.

[AM02] Abubakar, A, van den Berg, P.M. and Jordi J. Mallorqui. Imaging of biomedical data using a multiplicative regularized contrast source inversion method. *IEEE Transactions on Microwave Theory and Techniques*, 50(7):1761–1771, 2002.

[Amm08] H. Ammari. *An Introduction to Mathematics of Emerging Biomedical Imaging*. Springer, 2008.

[Ang90] G. Anger. *Inverse Problems in Differential Equations*. Akademieverlag, Berlin, 1990.

[AvdB00] A Abubakar and P.M. van den Berg. *A total variation enhanced contrast source inversion method for three-dimensional profile reconstruc-*

*tion*. European Congress on Computational Methods in Applied Sciences and Engineering, ECCOMAS, 2000.

[AvdB01] A. Abubakar and P.M. van den Berg. Total Variation as a Multiplicative Constraint for Solving Inverse Problems. *IEEE Transactions on Image Processing*, 10(9):1384–1392, 2001.

[AvdBHB03] A. Abubakar, P.M. van der Berg, T.M. Habashy, and H. Braunisch. Deblurring using Iterative Multiplicative Regularization Technique. *Proc SPIE, Medical Imagin 2003: Image Processing*, Vol.5032:pp.966–972, 2003.

[AvdBHB04] A. Abubakar, P.M. van den Berg, T.M. Habashy, and H. Braunisch. A Multiplicative Regularization Approach for Deblurring Problems. *IEEE Transactions on Image Processing*, 13(11):1524–1532, 2004.

[Bak84] A.B. Bakushinskii. Remarks on Choosing a Regularisation Parameter using the Quasi-Optimality and Ratio Criterion. *USSR Computational Mathematics and Mathematical Physics*, 24(4):181–182, 1984.

[BB18] M. Benning and M. Burger. Modern regularisation methods for inverse problems. *Acta Numerica*, pages 1–111, 2018.

[Blo81] F. Bloom. *Ill-Posed Problems for Integrodifferential Equations in Mechanics and Electromagnetic Theory*. SIAM, Philadelphia, 1981.

[BS93] C. Bouman and K. Sauer. A generalized Gaussian image model for edge-presering MAP estimation. *IEEE Transaction on Image Processing*, 2:296–310, July 1993.

[CABFB94] P. Charbonnier, G. Aubert, L. Blanc-Féraud, and M. Barlaud. Two deterministic half-quadratic regularization algorithms for computed imaging. *Proc. 1st IEEE ICIP, Austin, TX*, Nov 1994.

[CBFAB97] P. Charbonnier, L. Blanc-Féraud, G. Aubert, and M. Barlaud. Deterministic Edge-Preserving Regularization in Computed Imaging. *IEEE Transactions on Image Processing*, 6(2):298–311, 1997.

[CD11] G. Cloutier and F. Destrempes. Response to the Letter to the Editor-in-Chief on Manuscript Entitled: "A Critical Review and Uniformized Representation of Statistical Distributions Modeling the Ultrasound Echo Envelope". *Ultrasound in Medicine and Biology*, 37(4):675–676, 2011.

[CDlRS14] L. Calatroni, J. De los Reyes, and C.B. Schönlieb. Dynamic Sampling Schemes for Optimcal Noise Learning under Multiple Nonsmooth Constraints. *IFIP Advances in Information and Communication Technology*, 443, 2014.

[CDlRS17] L. Calatroni, J.C. De los Reyes, and C.B. Schönlieb. Infimal convolution of data discrepancies for mixed noise removal. *SIAM J.Image Sciences*, 10(3):1196–1233, 2017.

[Cha13] Damon M. Chandler. Seven Challenges in Image Quality Assessment: Past, Present, and Future Research. *ISRN Signal Processing*, 2013(Article ID 905685):53 pages, 2013.

[Che01] W. Cheney. *Analysis for Applied Mathematics*. Springer-Verlag New York Inc, 2001.

[CW79] P. Craven and G. Wahba. Smoothing noisy data with spline functions: estimating the correct degree of smoothing by the method of generalized cross-validation. *Numer.Math*, 31:377–403, 1979.

[DG94] V. Dutt and J.F. Greenleaf. Ultrasound echo envelope analysis using a homodyned K distribution signal model. *Ultrasonic Imaging*, 16(4):265–287, 1994.

[DHL+99] Y. Dai, J. Han, G. Liu, D. Sun, H. Yin, and Y-X. Yuan. Convergence properties of nonlinear conjugate gradient methods. *SIAM J. OPTIM*, 10(2):345–358, 1999.

[DHRC11] Y. Dong, M. Hintermüller, and M.M. Rincon-Camacho. Automated regularization parameter selection in multi-scale total vairation models for image restoration. *J Math Imaging Vis*, 40:82–104, 2011.

[DlR14] J.C. De los Reyes. Optimization of mixed variational inequalities arising in flow of viscoplastic materials. *Computational Optimization and Applications*, 52(3):757–784, 2014.

[EHN96] H.W. Engl, M. Hanke, and A. Neubauer. *Regularization of Inverse Problems*. Kluwer Academic Press, Dordrecht, 1996.

[EK05] H.W. Engl and P. Kügler. *Nonlinear Inverse Problems: Theoretical Aspects and Some Industrial Applications, in Multidisplinary Methods for Analysis, Optimization and Control of Complex Systems*. Capasso, V; Periaux,J. (Eds). Springer, 2005.

[EKN89] H.W. Engl, K. Kunisch, and A. Neubauer. Convergence rates for Tikhonov regularisation of nonlinear ill-posed problems. *Inverse Problems*, 5:523–540, 1989.

[Eva10] L.C. Evans. *Partial Differential Equations*. second ed., Graduate Studies in Mathematics, vol 19, American Mathematical Society, 2010.

[GB08] Michael Grant and Stephen Boyd. Graph implementations for nonsmooth convex programs. In V. Blondel, S. Boyd, and H. Kimura, editors, *Recent Advances in Learning and Control*, Lecture Notes in Control and Information Sciences, pages 95–110. Springer-Verlag Limited, 2008. `http://stanford.edu/~boyd/graph_dcp.html`.

[GB14] Michael Grant and Stephen Boyd. CVX: Matlab software for disciplined convex programming, version 2.1. `http://cvxr.com/cvx`, March 2014.

[GHW79] G.H. Golub, M. Heath, and G. Wahba. Generalized cross-validation as a method for choosing a good ridge parameter. *TECHNOMETRICS*, 21(2):215–223, 1979.

[Gla86] G. Gladwell. *Inverse Problems in Vibrations*. Nijhoff, Dordrecht, 1986.

[GM85] S. Geman and D.E. McClure. Bayesian image analysis:An application to single photon emission tomography. *Proc. Statistical Computation Section, Amer.Statistical Assoc.*, pages 12–18, 1985.

[Gol73] G.H. Golub. Some modified matrix eigenvalue problems. *SIAM Review*, 15:pp.318–334, 1973.

[GR92] S. Geman and G. Reynolds. Constrained restoration and the recovery of discontinuities. *IEEE Trans.Patterm Anal.Machine Intell.*, 14:367–383, Mar 1992.

[Gro77] C.W Groetsch. *Generalized Inverses of Linear Operators: Representation and Approximation*. Dekker, New York, 1977.

[HAH09] W. Hu, A. Abubakar, and T.M. Habashy. Simultaneous multifrequency inversion of full-waveform seismic data. *Geophysics*, 74(2):R1, 2009.

[Han92] P.C. Hansen. Analysis of discrete ill-posed problems by means of the L-curve. *SIAM Rev.*, 34:561–580, 1992.

[Han98] P.C. Hansen. *Rank-deficient and Discrete Ill-posed Problems: Numerical aspects of linear inversion*. SIAM, Philadelphia, 1998.

[Han09] P.C. Hansen. *Discrete Inverse Problems: Insight and Algorithms*. Society for Industrial and Applied Mathematics, Philadelphia, 2009.

[HN01] J.K. Hunter and B. Nachtergaele. *Applied Analysis*. World Scientific, 2001.

[HO93] P.C. Hansen and O'Leary. The use of the L-curve in the regularization of discrete ill-posed problems. *SIAM J.Sci.Comput*, 14:1487–1503, 1993.

[Idi13] J. Idier. *Bayesian Approach to Inverse Problems*. John Wiley & Sons, 2013.

[JP76] E. Jakeman and P. Pusey. A model for non-Rayleigh sea echo. *IEEE Transactions on Antennas and Propagation*, 24(6):806–814, 1976.

[Kir96] A Kirsch. *An Introduction to the Mathematical Theory of Inverse Problems*. Springer, 1996.

[KS08] B. Kaltenbacher, A. Neubauer and O. Scherzer. *Iterative Regularization Methods for Nonlinear Ill-Posed Problems*. Walter de Gruyter, Berlin, Germany, 2008.

[Kur98] K Kurdyka. On gradients of functions definable in o-minimal structures. *Ann. Inst. Fourier*, 48(3):796–783, 1998.

[Lam01] P.K Lamm. Some recent developments and open problems in solution methods for mathematical inverse problems. *Proc. 24 National Congress of Applied and Computational Mathematics (Belo Horizonte, Brazil, 10-13 Sept)*, Preprint, 2001.

[Lan90] K. Lange. Convergence of EM image reconstruction algorithms with Gibbs smoothing. *IEEE Trans.Med.Image.*, 9(4):439–446, Dec 1990.

[LH89] R. Leahy and T. Hebert. A generalized EM algorithm for 3-D Bayesian reconstruction from Poisson data using Gibbs priors. *IEEE Trans.Med.Image.*, 8(2):194–202, June 1989.

[LH13] A. Langer and M. Hintermüller. Subspace Correction Methods for a class of Nonsmooth and Nonadditive Convex Variational Problems with Mixed $L^1/L^2$ Data-Fidelity in Image Processing. *SIAM Journal on Imaging Sciences*, 6(4):2134–2173, 2013.

[Loj63] S. Lojasiewicz. *Une propriété topologique des sous-ensembles analytiques réels.* Les Équations aux Dérivées Partielles. Editions du centre National de la Recherche Scientifique, *Paris*, 1963.

[Loj93] S. Lojasiewicz. Sur la géométrie semi- et sous-analytique. *Ann. Inst. Fourier*, 43:1575–1595, 1993.

[Mor66] V.A. Morozov. On the solution of functional equations by the method of regularisation. *Soviet Math.Dokl.*, 7:414–417, 1966.

[Nas76] M.Z. Nashed. *Generalized Inverses and Applications.* Academic Press, New York, 1976.

[NC07] M. Nikolova and R.H. Chan. The equivalence of half-quadratic minimization and the gradient linearization iteration. *IEEE Trans.Image Processing*, 16(6):1623–1627, June 2007.

[NT17] M Nikolova and P Tan. Alternating proximal gradient descent for nonconvex regularised problems with multiconvex coupling terms. *HAL*, Id: hal-01492846, 2017.

[Pay75] L.E. Payne. *Improperly Posed Problems in Partial Differential Equations.* SIAM, Philadelphia, 1975.

[PT87] J. Pöschl and E. Trubowitz. *Inverse Spectral Theory.* Academic Press, Boston, 1987.

[R20a] Mathworks R2019a. Edge. `https://www.mathworks.com/help/images/ref/edge.html`. Accessed May 26. 2019.

[R20b] Mathworks R2019a. Medfilt2. `https://www.mathworks.com/help/images/ref/medfilt2.html`. Accessed May 26. 2019.

[Reg96] T. Regińska. A Regularization Parameter in Discrete Ill-Posed Problems. *SIAM J.Sci.Comput*, 17(3):740–749, 1996.

[Roc97] R. Rockafellar. *Convex Analysis*. Princeton University Press, 1997.

[Sch95] O Scherzer. Convergence Criteria of Iterative Methods Based on Landweber Iteration for Solving Nonlinear Problems. *J. Math. Anal. Appl.*, 194:911–933, 1995.

[She94] J.R. Shewchuk. An introduction to the conjugate gradient method without the agonizing pain. Technical report, CMU, Pittsburgh, PA, USA, 1994.

[Sho07] T.S. Shores. *Applied Linear Algebra and Matrix Analysis* . Springer, 2007.

[SS95] R.R. Schultz and R.L. Stevenson. Stochastic modelling and estimation of multispectral image data. *IEEE Trans.Image Processing*, 4:1109–1119, Aug. 1995.

[SSB06] H.R. Sheikh, M.F. Sabir, and A.C. Bovik. A Statistical Evaluation of Recent Full Reference Image Quality Assessment Algorithms. *IEEE Transactions on Image Processing*, 15(11):3440–3451, November 2006.

[Stu10] A.M. Stuart. Inverse Problems: A Bayesian Perspective. *Acta.Numerica*, 19:451–559, May 2010.

[SY06] W. Sun and Y-X. Yuan. *Optimization Theory and Methods, Nonlinear Programming* . Springer Optimisation and Its Applications, Vol.1, 2006.

[Vai82] G.M. Vainikko. The discrepancy principle for a class of regularisation methods. *USSR Comp.Math.Math.Phys.*, 22(3):1–19, 1982.

[Wah77] G. Wahba. Optimal Smoothing of density estimates. *Classification and Clustering*, pages 423–458, 1977.

[WW75a] G. Wahba and S. World. A completely automatic French curve: fitting spline functions by cross-validation. *Comm,in Statist.*, 4:1–17, 1975.

[WW75b] G. Wahba and S. World. Periodic splines for spectral density estimation: the use of cross-validation for determining the correct degree of smoothing. *Comm,in Statist.*, 4:125–141, 1975.

[WYY10] Y. Wang, C. Yang, and A.G. Yagola. *Optimization and Regularization for Computational Inverse Problems and Applications*. Springer, Berlin, Heidelberg, 2010.

[Zou70] G. Zoutendijk. *Nonlinear Programming, Computational Methods, in Integer and Nonlinear Programming*. J.Abadie, ed. North Holland Publishing Company, 1970.